



Universidad Nacional del Sur

TESIS DE DOCTOR EN CIENCIAS DE LA COMPUTACIÓN

Visualización Basada en Semántica

Martín Leonardo Larrea

BAHÍA BLANCA — ARGENTINA

2010



Universidad Nacional del Sur

TESIS DE DOCTOR EN CIENCIAS DE LA COMPUTACIÓN

Visualización Basada en Semántica

Martín Leonardo Larrea

BAHÍA BLANCA — ARGENTINA

2010

Prefacio

Esta Tesis es presentada como parte de los requisitos para optar al grado académico de Doctor en Ciencias de la Computación, de la Universidad Nacional del Sur, y no ha sido presentada previamente para la obtención de otro título en esta Universidad u otras. La misma contiene los resultados obtenidos en investigaciones llevadas a cabo en el Departamento de Ciencias e Ingeniería de la Computación, durante el período comprendido entre el 1 de abril de 2006 y el 17 de diciembre de 2010, bajo la dirección de la Dra. Silvia M. Castro, Profesora Titular del Departamento de Ciencias e Ingeniería de la Computación.

Martín Leonardo Larrea

`mll@cs.uns.edu.ar`

DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN

UNIVERSIDAD NACIONAL DEL SUR

Bahía Blanca, 17 de diciembre de 2010.



UNIVERSIDAD NACIONAL DEL SUR
Secretaría General de Posgrado y Educación Continua

La presente tesis ha sido aprobada el .../.../..., mereciendo

la calificación de(.....)

Agradecimientos

A todos, muchas gracias.

The Prestige

This thesis will be divided into three parts, “The Pledge”, “The Turn”, and “The Prestige”. The first part is called “The Pledge”; the posgrad student shows you something ordinary, but of course... it probably isn't. The posgrad student shows you a visualization process. Perhaps he asks you to inspect it to see if it is indeed real, unaltered, and normal. The second part is called “The Turn”; the student takes his ordinary visualization process and makes it do something extraordinary. The visualization process creates effective visualization without the user's participation; no parameters are set, no options are configured. Now you're looking for the secret... But you wouldn't clap yet. Because making effective visualizations isn't enough for a PhD; you have to show them how you did it. That's why this thesis has a third part, the hardest part, the part we call “The Prestige”. And in “The Prestige” you will see that's all based on semantic information. Based on Christopher Nolan's movie “The Prestige”

Resumen

El desafío de una visualización es encontrar una metáfora visual que permita entender y percibir en forma efectiva un conjunto de datos. Una visualización debe proveer también un conjunto de interacciones a partir de las cuales el usuario explorará el conjunto de datos con una mínima carga cognitiva. La tecnología computacional actual permite la exploración de grandes conjuntos de datos y espacios de información; dichos conjuntos están disponibles a través de diferentes sistemas de información. Por un lado, esta situación es extremadamente útil y excitante pero, la creciente cantidad de información genera una alta sobrecarga cognitiva llegando a causar ansiedad y estrés. Mientras que el poder computacional ha crecido en forma exponencial, la habilidad para interactuar con dichos datos sólo se ha incrementado en forma lineal. En la última década, el crecimiento exponencial del poder computacional ha permitido elaborar nuevas preguntas e intentar resolver problemas más complejos. La dimensionalidad de los datos también se ha incrementado, requiriendo mayores esfuerzos en identificar y comprender las relaciones relevantes para una tarea analítica particular.

Hoy en día, una gran variedad de usuarios acceden, extraen y muestran información que está distribuida sobre diferentes puntos, con diferentes tipos, formas y contenidos. En muchos casos, el usuario debe tener un control activo sobre el proceso de visualización pero, aún en este caso, se torna difícil obtener una visualización efectiva. Debido a que el objetivo de una visualización es lograr una representación que ayude al usuario a interpretar un conjunto de datos y comunicar su significado, es importante, por ejemplo, controlar el mapeo de las dimensiones físicas a las perceptuales; un usuario inexperto podría utilizar un mapeo incorrecto afectando negativamente la visualización resultante. Una estrategia para mejorar esta situación es guiar al usuario en la selección de los diferentes parámetros involucrados en la visualización.

Esta tesis tiene como objetivo crear un modelo de visualización que considere la semántica de los datos, del contexto y de las etapas del proceso de visualización para poder asistir al usuario en el seteo de los parámetros de la visualización para lograr un resultado efectivo. Este nuevo modelo transformará datos en información, de acuerdo a Keller & Tergan, "*information is data that has been given meaning through interpretation by way of relational connection and pragmatic context*". Los mismos datos pueden dar origen a diferente información, de acuerdo a la interpretación que se les de. Considerando esto, el nuevo proceso de visualización será capaz de determinar las características de una representación efectiva y podrá guiar al usuario a través de los diferentes estados del modelo.

En conclusión, consideramos que el modelado de un proceso de visualización junto con sus interacciones no es suficiente para asegurar una visualización efectiva. Para lograr esto debe ser desarrollado un meta modelo para el proceso de visualización, los estados, los datos, las interacciones y el contexto.

Abstract

The visualization challenge is to find a visual metaphor that the user can understand and perceive effectively combined with the interaction methods that make it possible for the user to work with and probe the data as effectively and effortlessly as possible. Computer technology allows the exploration of big information resources. Huge amount of data are becoming available on networked information systems, ranging from unstructured and multimedia documents to structured data stored in databases. On one side, this is extremely useful and exciting; on the other side, the ever growing amount of available information generates cognitive overload and even anxiety, especially in novice or occasional users. While computational power has increased exponentially, the ability to interact with useful information has only increased linearly. In recent decades, the exponential increase in computing power has allowed many more questions to be posed and more complex problems to be addressed. The dimensionality of data has also increased, requiring greater effort to identify and comprehend relationships relevant to a particular analytic task.

Now a day, a variety of users access, extract and display information that is distributed over different locations, with different types, forms and contents. In many cases, the user must have an active control over the visualization process, but even in this case, it becomes difficult to obtain an effective visualization. Because the goal of visualization is to make a representation that helps the user to interpret a set of data and communicate its meaning is important, for example, to control the mapping of the physical dimensions of the perceptual ones; an inexperienced user could use a incorrect mapping, negatively affecting the resulting visualization. A strategy to improve this situation is to guide the user in the selection of the various parameters involved in the process.

This thesis aims to create a visualization model that considers the semantics of data, context and visualization stages in order to assist the user in the settings of the

visualization parameters to achieve a successful result. This new model will transform data into information, according to Keller & Tergan “*information is data that has been given meaning through interpretation by way of relational connection and pragmatic context*”. The same data can be associated to different information, according to the interpretation of them. By considering this, the new visualization process will be able to determine the characteristics of an effective representation and will guide the user through the different states of the model.

In conclusion, we consider that the modeling of a visualization process together with their interactions is not sufficient to ensure an effective visualization. To achieve this goal a formal model for the visualization process, states, data, interactions and context must be developed.

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Contexto de la Tesis	3
1.3. Contribución de la Tesis	3
1.4. Estructura de la Tesis	6
2. Visualización	9
2.1. Automatización	10
2.2. Modelos de Visualización	21
2.2.1. Características del Proceso de Visualización	22
2.2.2. Modelo Unificado de Visualización	25
2.3. Taxonomías	32
3. Visualización y Semántica	37
3.1. Visualización Basada en Semántica	37
3.2. Semántica	37
3.2.1. Ontología	38
3.2.2. Description Logics	39
3.2.3. Ontology Web Language (OWL)	40
3.2.4. Razonador Semántico	41
3.3. Antecedentes	42
3.4. Semántica en el Proceso de Visualización	45

4. Caracterizaciones	49
4.1. Caracterización de los datos en Visualización	49
4.1.1. Antecedentes en caracterizaciones de datos	50
4.1.2. Nuestra propuesta para la caracterización de datos	66
4.1.3. Datos atómicos	68
4.1.4. Datos compuestos	71
4.1.5. Comparación de nuestra clasificación con los antecedentes presentados	73
4.2. Caracterización de las tareas en Visualización	76
4.2.1. Antecedentes en la caracterización de las tareas	76
4.2.2. Nuestra propuesta para la caracterización de tareas	81
4.3. Caracterización del contexto en Visualización	82
4.3.1. Antecedentes en la caracterización del contexto	82
4.3.2. Nuestra propuesta para la caracterización del contexto	84
4.4. Caracterización del humano en Visualización	88
4.4.1. Antecedentes en la caracterización del humano en Visualización . . .	88
4.4.2. Antecedentes en la caracterización del humano en la Web Semántica	89
4.4.3. Antecedentes en la caracterización del Humano en HCI	91
4.4.4. Nuestra propuesta para la caracterización del humano	97
4.5. Caracterización del proceso de visualización	102
4.5.1. Antecedentes en la caracterización del proceso de visualización . . .	102
4.5.2. Nuestra propuesta para la caracterización del proceso de visualización	102
5. Ontologías en el proceso de Visualización	103
5.1. Ontología del proceso de visualización	103
5.1.1. Ontología de Visualización	104
5.1.2. Procesamiento de datos	105
5.1.3. Representación Visual	107

5.1.4.	Proceso de visualización	109
5.1.5.	Extensiones	110
5.1.6.	Ontología de visualización y visualización basada en semántica . . .	112
5.2.	Ontología de los datos	113
5.3.	Ontología de las Tareas	115
5.4.	Ontología del Contexto	117
5.4.1.	Ontología del <i>Hardware</i>	118
5.4.2.	Ontología del Ambiente Físico	120
5.4.3.	Ontologías del Dominio de Aplicación y Dominio Organizacional . .	121
5.5.	Ontología del Usuario	121
5.5.1.	Ontología del Modelo Cultural	121
5.5.2.	Ontología del Modelo de Procesamiento de Información	121
5.6.	Ontología de Valoración	122
6.	Razonador Semántico en el Proceso de Visualización	127
6.1.	Los datos	127
6.1.1.	Nuevas relaciones	128
6.1.2.	Completar los datos	128
6.1.3.	Chequeo de consistencia	130
6.2.	Las tareas	131
6.3.	El Hardware	132
6.4.	La técnica de representación	133
6.5.	El mapeo visual	134
6.6.	La visualización	136
6.7.	El proceso de visualización	137

7. Casos de Estudio	139
7.1. Brows.AR: Visualizador de jerarquías de archivos	139
7.2. Primer Prototipo. Asignación de Color Basada en Semántica	140
7.2.1. Protégé, Jena & Pellet	141
7.2.2. Implementación	142
7.3. Segundo Prototipo. Creación de una Visualización utilizando Información Semántica	146
7.3.1. Ontología de Color	146
7.3.2. Ontología de Datos	146
7.3.3. Ontología de Tareas	147
7.3.4. Ontología de Hardware	148
7.3.5. Proceso de razonamiento	149
8. Conclusiones	155
9. Trabajo Futuro	159
9.1. Validación de los resultados	159
9.2. Prototipado	160
9.3. Líneas de investigación	160
A. Combinación de Colores	163
A.1. Evaluación con Usuarios	163
A.1.1. Técnica de Evaluación	163
A.1.2. Objetivos	164
A.1.3. Usuarios y Lugar	165
A.1.4. Resultados	166

Índice de figuras

1.1. Modelo de visualización distribuido y basado en semántica	4
1.2. Contexto de la tesis	5
2.1. Modelo lineal para la generación de representaciones	11
2.2. Ranking de precisión para tareas perceptuales por Cleveland y McGill . . .	12
2.3. Ranking de precisión de tareas perceptuales por Mackinlay	13
2.4. Mapeo de técnicas a sintaxis del lenguaje gráfico	14
2.5. Criterios de expresividad generales del lenguaje gráfico	15
2.6. Ejemplo de una respuesta dada por el sistema SAGE	17
2.7. Guías para la creación de mapas de colores en base a los tipos de datos . .	18
2.8. Características del Usuario	19
2.9. Características del Hardware	20
2.10. Tableau Software	21
2.11. El proceso de visualización	25
2.12. El Modelo Unificado de Visualización	26
2.13. El Modelo Unificado de Visualización en Visualización Científica.	35
2.14. El Modelo Unificado de Visualización en Visualización de Información. . .	36
3.1. Modelo de Representación Adaptable	43
3.2. Ejemplo de <i>Semantic Layers</i>	45
3.3. Elementos involucrados en el proceso de visualización	46

3.4. Semántica involucrada en el proceso de visualización	47
4.1. Tipos de datos descriptos por S. Stevens en 1946	50
4.2. Descripción de las características de un mapa de color	56
4.3. Caracterización de los datos introducida por Zhou y Feiner	60
4.4. Taxonomía de técnicas de visualización para un modelo continuo	64
4.5. Taxonomía de técnicas de visualización para un modelo discreto	64
4.6. Taxonomía de técnicas de visualización	65
4.7. Taxonomía de datos presentada por Pfitzner et al.	66
4.8. Taxonomía de relaciones por Pfitzner et al.	67
4.9. Nuestra propuesta para la clasificación de los datos	67
4.10. Se define cómo deben ser dos medidas M y M' para ser medidas válidas sobre cada tipo de escala.	68
4.11. Gráfico de relaciones entre instancias en la ontología de <i>FOAF</i>	89
4.12. Clases y propiedades utilizadas en <i>FOAF</i>	90
4.13. Descripción de una persona a través de la ontología <i>User Profile</i>	91
4.14. Descripción del contexto del usuario	92
4.15. Ontología basada en el modelo de perfil de usuario	93
4.16. Los elementos del humano	93
4.17. Modelo de Procesamiento de Información	95
4.18. Rueda de colores, con sus significados asociados según el modelo cultural.	101
5.1. Arquitectura de capas en la ontología de visualización	105
5.2. Vista de alto nivel de la ontología de visualización	106
5.3. Ejemplo de TBox para el caso de árboles de dialéctica en DeLP	106
5.4. Ejemplo de ABox para el caso de árboles de dialéctica en DeLP	107
5.5. Ejemplo de consultas en SparQL	108
5.6. Los espacios de color se representan como una instancia del concepto <i>Space</i>	109

5.7. Manejo de eventos en la ontología de visualización	110
5.8. Espacio de colores discretizado	112
5.9. Ontología de Color	113
5.10. Diagramado de la ubicación de la capa de inferencia	114
5.11. Diagrama de la ontología de datos propuesta	115
5.12. Representación del dato “Paciente”.	116
5.13. Ontología de Tareas.	117
5.14. Diagrama de la ontología de hardware	124
5.15. Ontología del ambiente físico	125
6.1. Uso del razonador semántico con los datos de entrada	128
6.2. Ejemplo sobre el cual se ejecutará el razonador	129
6.3. La relación descubierta por el razonador	130
6.4. Ejemplo sobre el cual ejecutará el razonador	131
6.5. Ejemplo de cómo el razonador puede completar datos	132
6.6. Ejemplo de una ontología inconsistente	133
6.7. Uso del razonador semántico con las tareas	134
6.8. Uso del razonador semántico con el <i>hardware</i>	134
6.9. Uso del razonador semántico con las técnicas de visualización	135
6.10. Uso del razonador semántico para la definición del mapeo visual	136
6.11. Uso del razonador semántico para la creación de la visualización	137
6.12. Uso del razonador a través del proceso de visualización	138
7.1. Captura de pantalla de Brows.AR	141
7.2. Diagrama de la ontología <i>VisualColorMapping</i>	143
7.3. Diagrama de la ontología <i>VisualColorMapping</i> para la representación de un tipo de archivo	144
7.4. Código principal del método <i>ask</i> en la clase <i>Reasoner</i>	145

7.5. Ontología de <i>Tarea</i>	148
7.6. Conjunto de reglas creadas que definen cómo se crea la visualización.	152
7.7. Arquitectura del sistema.	153
A.1. Rueda de colores discretizada	164
A.2. Una de las tarjetas de la “Evaluación de Combinación de Colores”	165
A.3. Tarjetas en blanco a ser completada por el usuario	166
A.4. Tarjeta usadas en la evaluación	167
A.5. Resultados de la evaluación	168
A.6. Nuevas tarjetas usadas en la segunda iteración de la evaluación	169
A.7. Resultados de la evaluación con los valores normalizados	170

Capítulo 1

Introducción

La esencia de la visualización es el uso de representaciones visuales e interacciones con el objetivo de obtener un rápido *insight* de un gran conjunto de datos. Card, Mackinlay y Shneiderman ([CMS99]) definen visualización como:

El uso de representaciones visuales interactivas de datos para ampliar la capacidad cognitiva.

Las representaciones visuales transforman datos en formas visuales que resaltan aspectos importantes, como anormalidades y similitudes en los datos. Estas representaciones visuales logran que el usuario perciba, en forma rápida, aspectos significativos de su conjunto de datos. Aumentando la capacidad cognitiva del usuario a través de representaciones visuales se logra un proceso analítico más rápido y enfocado. Sin duda es un gran reto crear representaciones visuales adecuadas. Cada representación visual debe permitir que el usuario explore los datos con efectividad. Tal exploración requiere que el usuario pueda interactuar con la representación y ganar *insight* a través de estas interacciones.

1.1. Motivación

El campo de la visualización ha madurado substancialmente durante la última década. Día a día el uso de visualizaciones se vuelve más generalizado, derivando en la necesidad de un entendimiento formal del proceso de visualización ([DBD04], [SAR08], [FMCL04]).

Los diferentes modelos de visualización presentados en los últimos años cubren en forma parcial el proceso de exploración; los modelos presentados por Upson ([UFK⁺89]) y Card ([CMS99]) proveen una vista general del proceso de visualización pero no modelan con suficiente detalle el modelo de exploración del usuario. El de Chi ([Chi00]) no hace una descripción apropiada de las interacciones y Chuah & Roth ([CR96]) presentaron un modelo con las interacciones muy detalladas pero no alcanzan a cubrir todos los posibles dominios de aplicaciones. Con el objetivo de cubrir las falencias de estos modelos y unificar sus virtudes se ha desarrollado un modelo que incluye todos los estados del proceso de visualización y las interacciones entre éstos y el usuario. El “Modelo Unificado de Visualización” ([MCFE03]) fue desarrollado para crear un marco de trabajo único y general, independiente del dominio de aplicación. El modelo se enfoca tanto en el proceso de visualización como en los estados de los datos.

Hoy en día, una gran variedad de usuarios acceden, extraen y muestran información que está distribuida sobre diferentes puntos, con diferentes tipos, formas y contenidos. En muchos casos, el usuario debe tener un control activo sobre el proceso de visualización pero, aún en este caso, es difícil obtener una visualización efectiva. Es común que la información que se desea representar no tenga una manifestación visual obvia; ante esta situación el proceso de mapeo del conjunto de datos a la vista puede llegar a ser no trivial ([Car03]). Debido a que el objetivo de una visualización es lograr una representación que ayude al usuario a interpretar un conjunto de datos y a comunicar su significado, es importante controlar el mapeo de las dimensiones físicas a las perceptuales; un usuario inexperto podría utilizar un mapeo incorrecto afectando negativamente la visualización resultante. El rol del usuario es primordial en el modelo de visualización debido a que es el/ella quien interactúa con la visualización y, basándose en su interpretación de la representación visual, modifica los parámetros de la visualización para ajustarla a su objetivo.

El espacio de los parámetros es normalmente muy grande y, cuando se manejan grandes conjuntos de datos, las interacciones del usuario pueden ser muy lentas. En la última década se ha trabajado mucho para mejorar la velocidad de las herramientas de visualización de modo que los usuarios pudieran realizar sus interacciones a velocidades interactivas, explorar espacios de parámetros más grandes y, con suerte, encontrar resultados satisfactorios más rápidamente. Sin embargo, con el crecimiento tanto de la cantidad de datos como de las técnicas de visualizaciones disponibles, la búsqueda de

una visualización adecuada también es más amplia. Por otro lado, en un proceso de visualización, el conocimiento del usuario es una parte indispensable de la visualización; sin embargo, la falta de cierto conocimiento por parte del usuario es a menudo un obstáculo para lograr que la visualización revele información significativa. Ambos escenarios sugieren la necesidad de asistir al usuario en la creación de la visualización y una manera de hacerlo es guiándolo. Una situación ideal sería poder guiar al usuario en la toma de decisiones durante el proceso de visualización de modo tal de lograr una visualización efectiva. Dicho resultado se obtendría aún si el usuario no tiene los conocimientos necesarios para lograrlo por si solo.

1.2. Contexto de la Tesis

En el Laboratorio de Investigación y Desarrollo en Visualización y Computación Gráfica (VyGLab) se ha definido un modelo de visualización unificado (MUV) que brinda tanto al diseñador como al usuario una guía en el proceso de visualización ([MCFE03]). A partir de este modelo, uno de los objetivos del VyGLab es el diseño y el desarrollo de un modelo de visualización distribuido, que extendiese al MUV y que esté basado en información semántica (Figura 1.1). Diferentes miembros del VyGLab se encuentran actualmente trabajando en diferentes partes del modelo de visualización (Figura 1.2). La formalización del MUV y su traducción en ontologías está siendo desarrollada por el Lic. Ing. Sebastián Escarza. La transformación del MUV en un modelo de visualización distribuido está a cargo de la Ing. Luján Ganuza. La caracterización de los datos de entrada y la incorporación de diagramados 3D es responsabilidad de la Ing. Dana Urribarri y el Mg. Martín Larrea, ambos aportes desarrollados como extensiones a la formalización del modelo. La capa de inferencia, que permitirá asistir al usuario en la toma de decisiones, está a cargo del Mg. Martín Larrea y es el tema central de esta tesis. Esta tesis también incluirá extensiones sobre la formalización del modelo.

1.3. Contribución de la Tesis

Dado un determinado conjunto de datos, un usuario toma decisiones acerca de qué herramientas de visualización debe usar para explorarlo y obtener *insight* de ese conjunto. Es en este proceso que utiliza diferentes técnicas, mapeos, espacios, etc. hasta

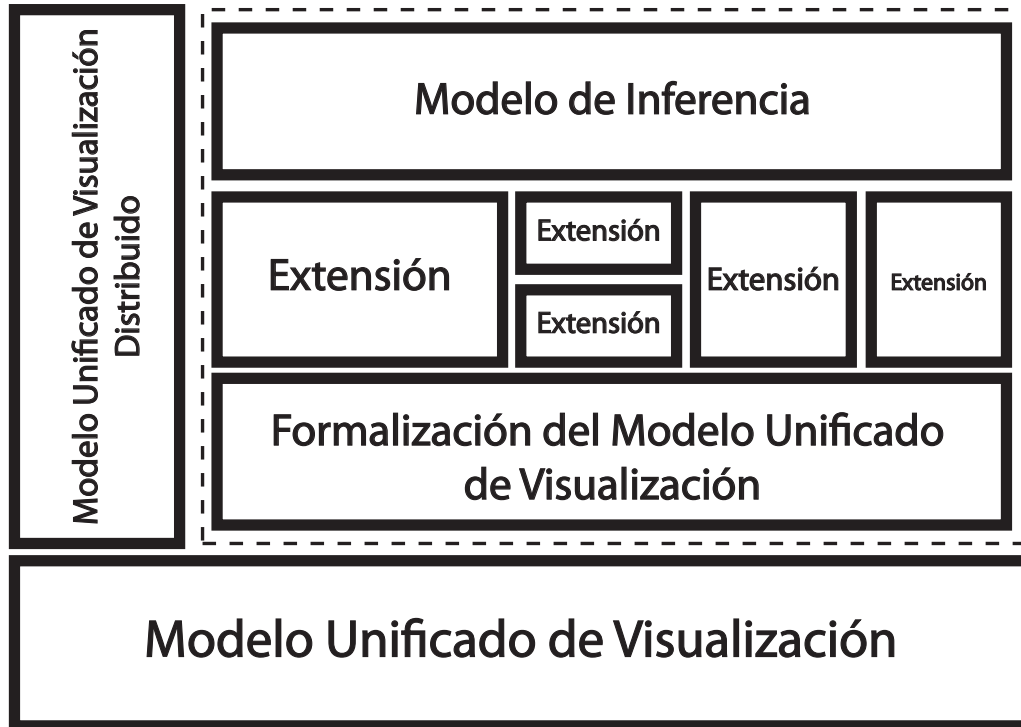


Figura 1.1: El VyGLab tiene como objetivo diseñar y desarrollar un modelo de visualización, que extendiese al MUV, basado en información semántica y distribuido.

lograr una visualización satisfactoria. Este proceso puede considerarse un proceso típico de búsqueda en un espacio sumamente complejo ([CEH⁺09]). En esta búsqueda, el espacio de los parámetros es excesivamente grande (por ejemplo, explorar distintos mapas de colores, distintos puntos de vistas y/o distintas funciones transferencia) y la misma resulta complicada. Una estrategia para mejorar esta situación es guiar al usuario en la selección de los diferentes parámetros involucrados en la visualización a través de información semántica. La Web Semántica es una extensión de la actual World Wide Web que se basa en el intercambio de información con una descripción explícita, formal y *machine-accessible* de significado ([HF01]). El desarrollo de la Web Semántica hizo posible la existencia de las herramientas necesarias para que una computadora pudiera entender y manipular información semántica ([SEZJM08]). El concepto de información semántica y la existencia de dichas herramientas para su manipulación y entendimiento fueron parte de la motivación para el desarrollo de este trabajo de tesis doctoral.

La contribución de esta tesis la constituye la creación de un modelo de visualización que

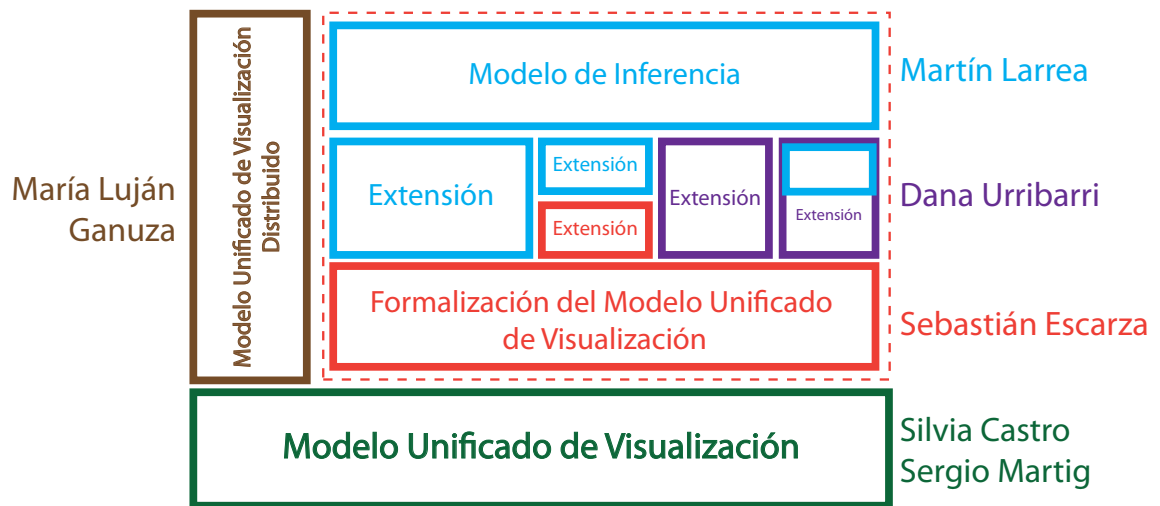


Figura 1.2: Diferentes miembros del VyGLab se encuentran actualmente trabajando en diferentes partes del modelo de visualización. Esta tesis se enfocará principalmente en el “Modelo de Inferencia”.

considere la semántica de los datos y de las etapas del “Modelo Unificado de Visualización” para poder asistir al usuario en el seteo de los parámetros de la visualización permitiéndole así lograr un resultado efectivo, independientemente de los conocimientos que tenga éste en el área de visualización. Este nuevo modelo transformará datos en información, de acuerdo a Keller & Tergan , *“information is data that has been given meaning through interpretation by way of relational connection and pragmatic context”*. Un mismo conjunto de datos puede dar origen a diferente información, de acuerdo a la semántica que se les de. Considerando esto, el nuevo proceso de visualización permitirá determinar las características de una representación efectiva y podrá guiar al usuario a través de los diferentes estados del modelo. Si bien ésta es la contribución principal de la tesis, no es la única; los aportes realizados en esta tesis son:

- La definición de un modelo de visualización basado en semántica
- Una caracterización/ontología de los datos a visualizar.
- Una caracterización/ontología de las tareas a realizar sobre una visualización
- Una caracterización/ontología del humano, considerando al humano como desarrollador de la visualización y como usuario o consumidor de la misma. Aquí nos

enfocamos tanto en la descripción del humano a partir de su información personal (*nombre, edad, lugar de nacimiento*) como también desde un punto de vista psicofísico (*memoria visual, reconocimiento de color*). Un tercer ítem que se incluye como parte de este trabajo es la caracterización del modelo cultural asociado al usuario.

- Una caracterización del contexto en el que ocurre la visualización. Por contexto consideramos el ambiente físico en el cual está trabajando el humano (*nivel de luz, de ruido*), el hardware que utiliza (*tipo de cpu, tipo de gpu, tamaño de monitor*), el contexto como dominio de aplicación en el que se encuadra la visualización (*ingeniería eléctrica, medicina, geología*) y finalmente el dominio organizacional.

1.4. Estructura de la Tesis

En esta tesis incluimos los conceptos relevantes, de modo tal que la lectura de la misma sea autocontenida. A continuación describimos su estructura:

Capítulo 1 Introducción. Se introducen los objetivos y el marco en el que se desarrolla esta investigación. También se enumeran cuáles son las contribuciones de la misma.

Capítulo 2 Visualización. En este capítulo se introduce al área de visualización, detallando los campos que cubre esta disciplina. Se analiza la creación de visualizaciones como un proceso constituido por estados y transformaciones y se detallan los diferentes modelos presentados que buscan formalizar el proceso de visualización.

Capítulo 3 Visualización y Semántica. En los últimos años han surgido ejemplos de integración de información semántica en visualización. Tales trabajos son descriptos en este capítulo, analizando cómo se hizo dicha integración, sus ventajas y desventajas.

Capítulo 4 Caracterizaciones. Existen tres niveles de formalismo: terminología, taxonomía o caracterización y ontología ([DBDH05]). Para obtener una visualización basada en semántica es necesario contar con ontologías que den soporte a la semántica. Éstas se pueden obtener a partir de las caracterizaciones existentes de los

elementos involucrados en el proceso de visualización, por ejemplo los tipos de datos. En este capítulo se presenta una evaluación de las caracterizaciones existentes, y se describen detalladamente nuestras propuestas.

Capítulo 5 Representación de las caracterizaciones. En este capítulo se presentan las ontologías que se crearon para dar soporte a las caracterizaciones propuestas en el capítulo anterior.

Capítulo 6 Razonador Semántico en el Proceso de Visualización. Una de las ventajas de utilizar semántica en el proceso de visualización, es la de derivar nueva información a partir de un razonador semántico. Una de las contribuciones de esta tesis es la incorporación del razonador semántico al proceso de visualización. En este capítulo se explica cómo dicho razonador será utilizado.

Capítulo 7 Casos de Estudio. Con el objetivo de validar el modelo teórico presentado se realizaron diferentes implementaciones del mismo. Este capítulo detalla los prototipos implementados, siendo cada uno de estos variantes de una aplicación para la visualización de estructuras de directorios.

Capítulo 8 Conclusiones. Aquí se presentan las conclusiones de nuestro trabajo.

Capítulo 9 Trabajo Futuro. Son múltiples las líneas de investigación que se pueden establecer a partir de este trabajo. En este capítulo final se define el trabajo futuro a realizar.

Apéndice A Evaluación sobre la combinación de colores. Al definir la semántica de los colores se han establecido diferentes relaciones entre ellos. Para validar dichas relaciones se ha realizado una evaluación con usuarios. En éste apéndice se describe dicha evaluación.

Capítulo 2

Visualización

La visualización es el proceso por el cual se crea una representación visual interactiva a partir de un conjunto de datos. Una visualización exitosa provee una representación que permite ganar *insight* de los datos o comunicar aspectos de los mismos de manera efectiva ([Ber83], [Tuf86], [Cle85]). La visualización es clave en las ciencias y las ingenierías y es utilizada con amplios propósitos. De acuerdo a Bergeron ([BAB⁺93]), “una visualización tiene como objetivo describir, analizar y explorar un conjunto de datos”.

Este capítulo se enfocará en el trabajo realizado en el área de Visualización que sirvió de base para el desarrollo de esta tesis. El trabajo previo ha sido ordenado en tres secciones:

1. **Automatización.** En esta sección se detallan los trabajos y avances realizados en la automatización del proceso de visualización realizado hasta el momento.
2. **Modelos de Visualización.** Aquí se describen los modelos que formalizaron, con diferentes grados de formalismo, el proceso de visualización y que tienen un vínculo significativo con el trabajo realizado en esta tesis.
3. **Taxonomías.** Para poder comprender y analizar los tipos de datos, las técnicas y las tareas involucradas en el proceso de visualización se han desarrollado clasificaciones y taxonomías de las mismas. En el contexto de la definición de una semántica de visualización, dichas clasificaciones y taxonomías son sumamente importantes. En esta sección presentaremos aquéllas que consideramos relevantes para el desarrollo de este trabajo.

2.1. Automatización

La visualización, como la conocemos hoy, tuvo sus inicios a finales de los '40. Desde ese momento hasta la actualidad se puede apreciar que para obtener una visualización efectiva, el proceso de visualización requiere del usuario conocimientos específicos de áreas como diseño gráfico, psicología, interacción humano-computadora, etc. No sólo son necesarios estos conocimientos, sino también la capacidad de combinarlos en forma adecuada.

En 1994, Roth et. al ([RKMG94]) detectaron este problema e identificaron tres limitaciones en los paquetes gráficos del momento que, en diferente medida, aún son válidas hasta el día de hoy:

1. No permiten integrar vistas para ver las relaciones entre distintos conjuntos de datos.
2. Tienen interfaces gráficas complejas.
3. No guían al usuario en el proceso de creación de visualizaciones. La calidad de la visualización depende de los conocimientos y habilidades del usuario en el área.

Los problemas detectados, particularmente el detallado en el *ítem 3*, presentan una clara intersección con la motivación de esta tesis. Se puede ver entonces que dichos problemas han estado presentes entre nosotros, e identificados, desde hace mucho tiempo y sin embargo aún no han sido solucionados exitosamente.

En 1986 Jock Mackinlay estableció como objetivo el desarrollo de una aplicación para automatizar la creación de representaciones gráficas ([Mac86]). Según este autor, el usuario no siempre tiene el conocimiento necesario para crear una buena representación. Mackinlay se basó en la idea de que una representación gráfica es una expresión en un lenguaje gráfico, donde ese lenguaje se encuentra definido formalmente. Si bien Mackinlay estableció limitaciones al alcance de su trabajo, sólo representaciones 2D, es la primera referencia en cuanto a la creación de un proceso automático de visualización. Como se podrá apreciar al concluir esta sección, los trabajos desarrollados en los últimos años sobre la automatización de este proceso, toman como base fundamental el trabajo presentado por Mackinlay.

Mackinlay pensó un sistema, al cual denominó *Presentation Tool*, que pudiese sintetizar un diseño gráfico y renderizar una imagen a partir de un conjunto de datos de entrada (Figura 2.1). En el contexto de su trabajo, un diseño gráfico es una descripción

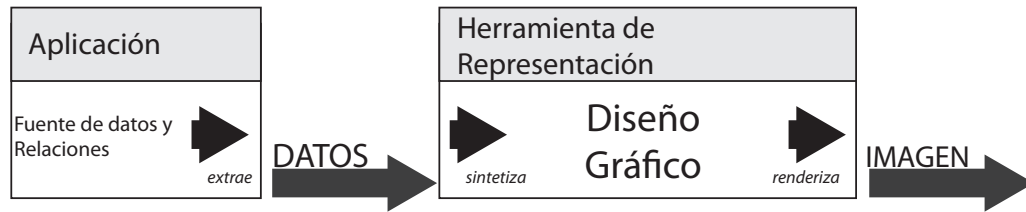


Figura 2.1: Modelo lineal para la generación de representaciones ([Mac86]). Un diseño gráfico es sintetizado por la aplicación para crear las representaciones y a su vez es utilizado para la renderización de la visualización.

abstracta, una indicación de las técnicas, variaciones de color, posición en el eje, etc., que se utilizaron para codificar la información.

El usuario no siempre tiene el conocimiento necesario para crear una buena representación; sin embargo, esta afirmación plantea dos grandes inquietudes:

- ¿Qué es una buena representación y cómo se sabe si ésta es efectiva o no?
- Si se tuviese una forma de saber qué es una buena representación, ¿cómo se codifican dichas características para ser procesadas por un sistema en forma automática?

Para responder a la primera pregunta, Mackinlay se basó en el trabajo realizado por Cleveland y McGill ([CM84]). En dicho trabajo se establece un ranking de tareas perceptuales (Figura 2.2); este ranking se ordena en función de la precisión con la que se puede diferenciar la información representada: a mayor nivel en el ranking, mayor precisión. De esta forma, Cleveland y McGill reconocen que un usuario podrá diferenciar con mayor precisión información representada mediante posiciones que mediante color.

Mackinlay reconoció la validez del ranking presentado pero a su vez señaló que dicho ranking variaba de acuerdo al tipo de dato que se quería representar. Es por esto que Mackinlay creó tres rankings, uno para cada tipo de dato, cuantitativo, ordinal y nominal (Figura 2.3). Utilizando estos nuevos rankings Mackinlay obtuvo una mejor herramienta para poder comparar y evaluar una representación.

Para poder formalizar esta forma de comparar representaciones Mackinlay definió una métrica denominada **efectividad**:

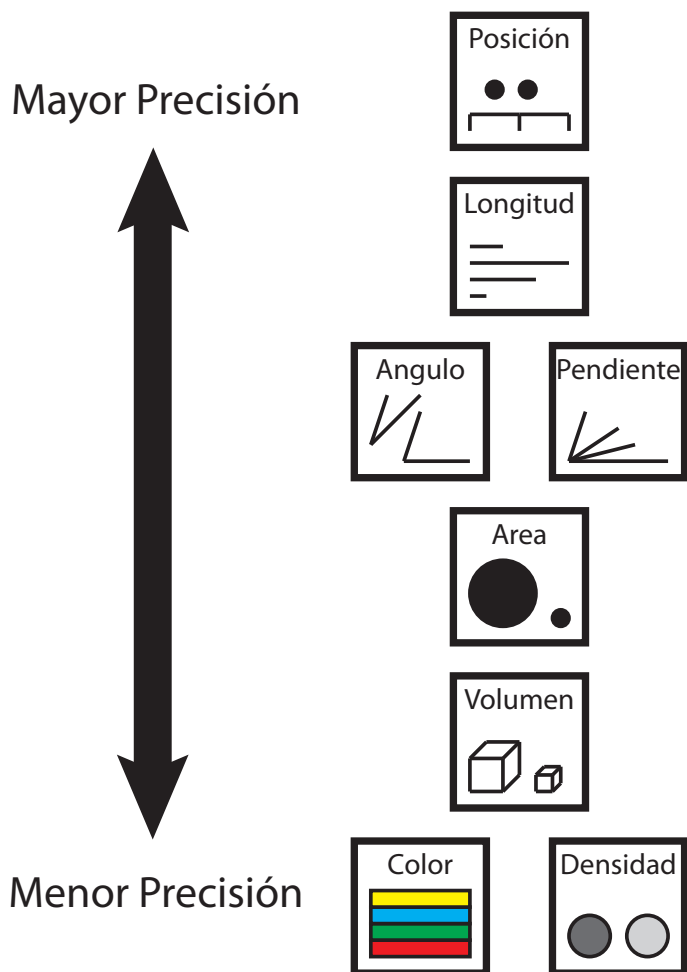


Figura 2.2: Ranking de precisión para tareas perceptuales presentado por Cleveland y McGill ([CM84]).

- **Effectivity:** *Identify which of a serie of graphical languages, in a given situation, is the most effective at exploiting the capabilities of the output medium and the human visual system ([Mac86]).*

Mackinlay reconoce que esta métrica es una medida muy relativa. Puede argumentarse que una representación es efectiva si es renderizada rápidamente o con pocos recursos. Mackinlay tomó la decisión de considerar que una representación es efectiva cuando puede ser interpretada en forma precisa. De la definición presentada es importante destacar que la métrica de efectividad no depende solamente de la representación visual sino también de los siguientes factores:

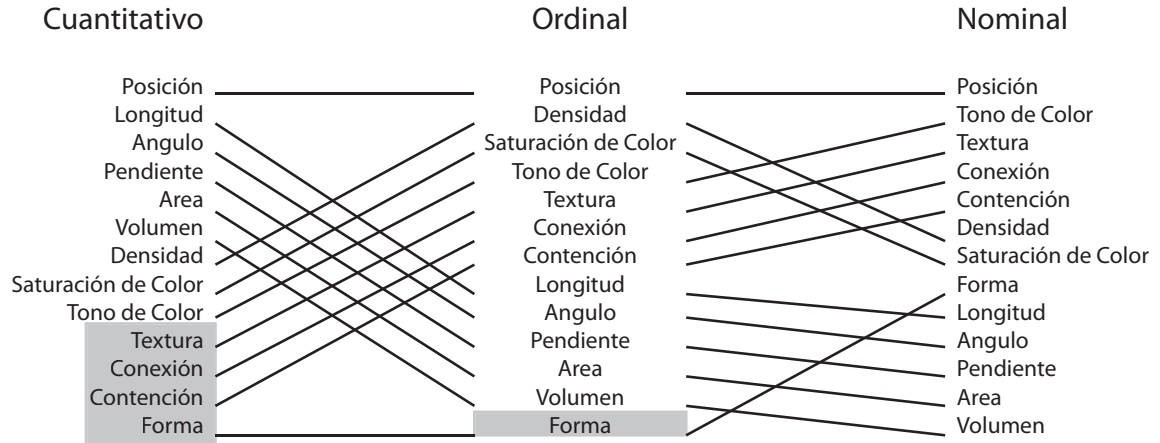


Figura 2.3: Ranking de precisión para tareas perceptuales presentado por Mackinlay ([Mac86]). Las palabras sombreadas son aquellas que no se aplican para el tipo de dato en particular.

- La situación en la que ocurre la representación. Mackinlay escribió *in a given situation* destacando la importancia del contexto en el que el usuario interactúa con la representación.
- El medio sobre el cual se realiza la representación. La definición de la métrica incluye la siguiente expresión: *the capabilities of the output medium* indicando la importancia que tiene el conocer las características del medio de salida.
- La importancia del humano y sus características al momento de evaluar una representación. Este reconocimiento se observa en la definición de efectividad al incluir las capacidades del sistema visual del humano, *at exploiting the capabilities of the... and the human visual system*.

Al observar la definición de efectividad podemos ver cómo Mackinlay establece la necesidad de caracterizar al **usuario**, que es quien va a utilizar la visualización, a las **tareas**, que se llevarán a cabo con la visualización, y al **contexto**, que constituye el ambiente en el cual ocurre la visualización. Estos tres elementos, conocidos como tríada de caracterizaciones, se continuaron estudiando y hoy en día son una pieza fundamental en el área de Interacción Humano-Computadora ([BGBG95]).

El segundo problema que tuvo que resolver Mackinlay fue el de cómo decidir codificar estas reglas de representación para que pudieran ser utilizadas por un

sistema automático. Enfocándose solamente en representaciones 2D de datos relacionales, Mackinlay reconoció la necesidad de contar con un lenguaje formal que describiese las propiedades sintácticas de una representación. Sin embargo, en lugar de utilizar un solo lenguaje de alta complejidad, definió una serie de lenguajes más simples (Figuras 2.1 y 2.1) con objetivos más específicos y un conjunto de primitivas a través de las cuales pudiesen ser combinados. Para ello definió una serie de operadores de composición utilizando un álgebra de composición.

En este caso Mackinlay definió una métrica adicional denominada **expresividad**, aplicable sobre un lenguaje, de la siguiente manera:

- **Expressivity:** *Identify graphical languages that express the desired information.*

La expresividad se asocia a cada lenguaje; cuando un lenguaje incluye toda y solamente toda la información que se quiere representar se dice que cumple con el criterio de expresividad. Un lenguaje que agregue información que el usuario no indicó que quiere representar o que quitase información que el usuario quiere representar se dice que no cumple con este criterio.

Técnicas de Codificación	Primitiva Gráfica del Lenguaje
Single Position	Eje Horizontal, Eje Vertical
Apposed Position	Gráfico de Líneas, Gráfico de Barras, Plot Chart
Retinal List	Color, Forma, Tamaño, Saturación, Textura, Orientación
Map	Mapa de Caminos, Mapa Topográfico
Connection	Arboles, Grafo Acíclico, Redes
Misc. (angle, contain, ...)	Gráfico de Torta, Diagramas de Venn, ...

(a) Conjunto de primitivas para el lenguaje gráfico

Técnicas de Codificación	Estructura Sintáctica
Single Position	$h(m), v(m)$
Apposed Position	$vh(m)$
Retinal List	m
Map	$vh(m)$
Connection	$m_i(m1)$
Misc. (angle, contain, ...)	$vh(m)$

(b) Estructura sintáctica de las primitivas

Figura 2.4: Mapeo de técnicas a sintaxis del lenguaje gráfico

Una vez definidas las métricas de **expresividad** y **efectividad** Mackinlay consideró que contaba con las herramientas necesarias para definir su proceso automático.

Técnicas de Codificación	Criterios de Expresividad
Single Position	$X \rightarrow Y$ (X es nominal)
Apposed Position	$X \times Y$ (X, Y no son nominales)
Retinal List	$X, o X \rightarrow Y$ (X no es cuantitativo)
Map	$L \rightarrow X, \dots$ (L es una ubicación)
Connection	$X \times X$ (X es nominal)
Misc. (angle, contain, ...)	Generalmente, $X \times Y$

(a) Criterios de expresividad generales para las primitivas del lenguaje, criterios más específicos pueden ser definidos.

	Nominal	Ordinal	Cuantitativo
Tamaño	-	*	*
Saturación	-	*	*
Textura	*	*	
Color	*	*	
Orientación	*		
Forma	*		

(b) Expresividad de técnicas retinales.

Figura 2.5: Criterios de expresividad generales del lenguaje gráfico

Como se observa en la Figura 2.1, el proceso definido por Mackinlay toma como entrada un conjunto de datos y genera una representación. En dicho proceso se crea un diseño gráfico. Un diseño gráfico, como se mencionó anteriormente, es una descripción abstracta de una representación y se encuentra codificado utilizando los lenguajes formales definidos por Mackinlay.

Finalmente, Mackinlay concluyó que se debe conocer la importancia de las relaciones que se quieren representar para saber cuáles hay que priorizar. Como él establece “*the input to the presentation tool is actually a tuple of relations that indicates the relative importance of the relations*”. Por ejemplo, si el usuario indica que lo más importante a representar es el elemento *país* cuyo tipo de dato es nominal entonces, de acuerdo a los rankings definidos, este elemento será representado por medio de posiciones.

Aunque Mackinlay incluye las características del usuario, del medio y de las tareas como factores importantes al momento de definir si una representación es efectiva o no, en ningún momento considera estos elementos como parte de su sistema automático. Estos factores recién se comenzaron a incluir en trabajos posteriores.

Los criterios que definió Mackinlay, **efectividad** y **expresividad**, son usados por Roth en 1991 ([RM91]) y en 1994 ([RKM94]) en SAGE, un sistema automático para la creación de representaciones que incluye el sistema de Mackinlay. Lo más destacable del

trabajo de Roth, en el contexto de esta tesis, es la inclusión de los objetivos del usuario como una de las entradas al sistema. El sistema que presenta Roth toma como datos de entrada los datos que el usuario quiere representar, sus características y los objetivos del usuario. Roth incorpora al trabajo de Mackinlay una nueva condición sobre la cual ordenar las representaciones. Mackinlay ordenaba en base al criterio de efectividad considerando los rankings presentados anteriormente; Roth redefinió el criterio de efectividad para que incluyese también los objetivos del usuario.

Roth señala que, “... *a presentation system must also consider the role of an application’s or user’s goals in viewing data...*”. De acuerdo a su trabajo presenta seis posibles tareas que el usuario puede plantear sobre una representación:

- Búsqueda precisa de valores (*accurate value lookup*).
- Comparación de valores dentro de una relación y no entre relaciones (*comparison of values within, but not between different relations*).
- Comparación de relaciones dentro del mismo conjunto de datos (*pairwise or n-wise comparison of relations for the same dataset*).
- Distribución de los valores en una relación (*distributions of values for a relation*).
- Las correlaciones entre los atributos (*correlations among attributes*).
- Localización de datos en función de un objeto (*data-locating needs for the objects referred to by a relation*).

La Figura 2.6 muestra un ejemplo del sistema SAGE de Roth en donde se establecen las relaciones, los objetivos y las tareas que el usuario desea representar.

Tanto Mackinlay como Roth parten del trabajo realizado por McGill y Cleveland ([CM84]) al momento de definir un ranking de tareas perceptuales. Se puede apreciar en la Figura 2.2 que el color quedó relegado a último lugar; Mackinlay reconoce que el color es muy efectivo para codificar conjuntos nominales y lo utiliza en su sistema como una forma válida y efectiva de representar información. Sin embargo no establece, y tampoco lo hace Roth, una relación entre las características del color y las características de los datos que representan.

Show Relations:
 Has-part
 Responsible-for
 Requires-resource

For Objects:
 Departments
 Departments, Activities
 Activities, Resources

Presentation Goals:
 Lookup-accurately - Has-part, Responsible-for,
 Requires-resource
 Locate-easily - Resources, Departments

SAGE's Response:

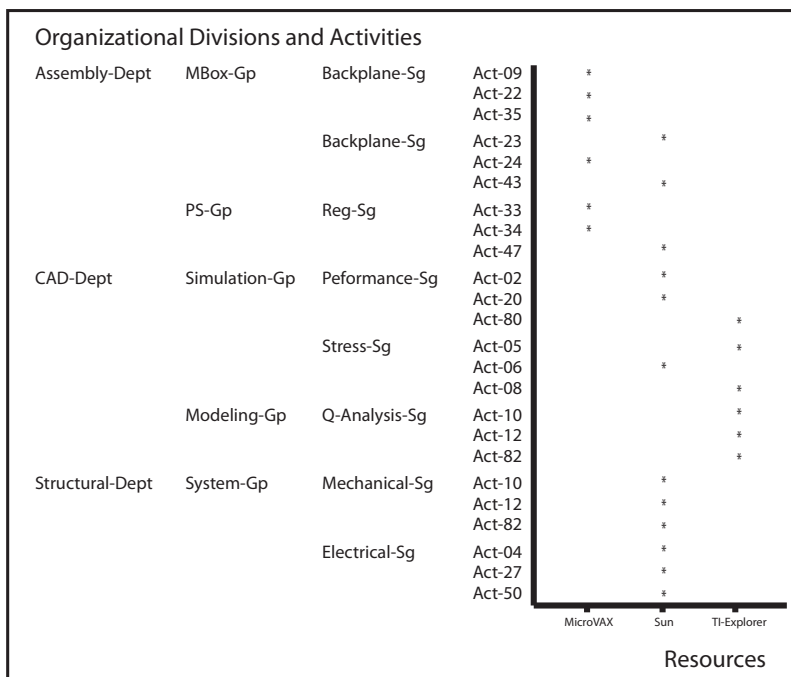


Figura 2.6: Respuesta del sistema SAGE a la entrada dada ([RKMG94]).

En 1995, investigadores de los laboratorios de IBM presentaron ([BRT95]) una serie de reglas que relacionan las características de los datos, llamados metadatos, con las características del mapa de color que los representan.

Las características que se evalúan en este trabajo son: metadatos provistos por el sistema, metadatos calculados y metadatos provistos por el usuario; además, se incluyen los tipos de datos junto a sus rangos; también son consideradas las tareas a desarrollar sobre la representación. Los tipos de datos soportados son: razón, intervalo, ordinales y

nominales; la frecuencia espacial se divide en baja o alta y las tareas a desarrollar son tres: isomorfismo, segmentación y resaltado. La Figura 2.7 muestra la tabla creada para tal objetivo.

Tipo de dato	Frecuencia Espacial	Tareas		
		<i>Isomorphic</i>	Segmentación	Resaltado
RATIO	Baja	Luminancia: Uniforme. Tono: Pares opuestos o complementarios. Saturación: Monotonicamente creciente desde el gris.	Número de segmentos par. Posiblemente muchos segmentos.	Rango amplio para permitir el resaltado
	Alta	Luminancia: monotonicamente creciente. Tono: Opuestos o complementarios. Saturación: Monotonicamente creciente desde el gris.	Número de segmentos par. Posiblemente pocos segmentos.	Rangos pequeños para permitir el resaltado.
INTERVALO	Baja	Luminancia: Uniforme. Tono: Pares opuestos. Saturación: Monotonicamente creciente desde el gris.	Posiblemente muchos segmentos.	Rango amplio para permitir el resaltado.
	Alta	Luminancia: Monotonicamente creciente desde el gris. Tono: Uniforme o con pequeñas variaciones de tono. Saturación: Monotonicamente decreciente.	Posiblemente pocos segmentos.	Rango pequeño para permitir el resaltado.
ORDINAL	Baja	Luminancia: Uniforme. Tono: Variaciones sobre el disco de tonos. Saturación: Monotonicamente decreciente.	Posiblemente pocos segmentos.	Incremento de luminancia en el área resaltada.
	Alta	Luminancia: Monotonicamente creciente. Tono: Variaciones sobre el disco de tonos. Saturación: Uniforme.	Posiblemente muchos segmentos.	Incremento de saturación en el área resaltada.
NOMINAL	Baja	Luminancia: Uniforme. Tono: Variaciones sobre el disco de tonos.	Menos de 7 segmentos.	Incremento de luminancia o saturación en el área resaltada.
	Alta	Saturación: Uniforme.		

Figura 2.7: Guías para la creación de mapas de color en base a los tipos de datos ([BRT95]).

Es importante mencionar que en este trabajo las características de los datos definen las características de los colores, sin embargo nunca se definen reglas para asociar un dato con un color o rango de colores específico, por ejemplo relacionar un dato con el color rojo. En muchos casos, por cuestiones culturales o por el tipo de dato que se desea representar, es natural asociar un color específico con un dato. Por ejemplo, azul con agua, rojo con peligro y verde con naturaleza. Otro elemento ausente en este trabajo es la definición de reglas para las relaciones entre colores; si las características de un dato definen las

características del color que lo representa entonces, las características de las relaciones entre los datos deberían reflejarse en las características de las relaciones entre los colores que los representan. Si dos datos están representando información opuesta, entonces los colores que los representan también deberían transmitir la misma información.

Si retomamos el trabajo de Mackinlay, podemos recordar que él consideró como parte de la métrica de efectividad las características del usuario y del medio en donde se presenta la visualización. Sin embargo, en su sistema automático no incluyó estas características. Golemati et al. ([GHV⁺06]) presentan por primera vez las características del usuario y del medio como elementos clave para la configuración de una visualización.

En este trabajo se consideraron las características del usuario (Figura 2.8) y del hardware que se está utilizando (Figura 2.9). En función de estos valores se configura la visualización.

Propiedades del contexto del usuario	
Propiedades del Contexto	Valores
Educación	Primaria Secundaria De grado
Título, rol o relación con la universidad	Miembro de la universidad Personal administrativo Estudiante Ninguna de las anteriores
Capacidad para la recuperación de información	Perfecta Media Baja
Objetivo	Investigación Publicación Información personal
Información general	Edad Género Profesión
Habilidades	Memoria visual Memoria aritmética Reconocimiento de color

Figura 2.8: Síntesis de las características del usuario ([GHV⁺06]).

Entre 2007 y 2008 se presentaron los trabajos “A Model and Framework for Visualization Exploration” ([JKMG07]), “Querying and Creating Visualization by Analogy” ([SVK07]) y “VisComplete: Automating Suggestions for Visualization Pipelines” ([Koo08]). Estos trabajos tienen en común la estrategia que utilizan para

Propiedades del contexto de <i>hardware</i>	
Propiedades del Contexto	Valores
Dispositivos de entrada	Mouse
	Teclado
	Joystick
	Dispositivos especializados
Dispositivos de salida	Monitor 2D
	Monitor 3D
	Dispositivo <i>Head-mounted</i>
Otro equipamiento de hardware	Procesador
	Memoria
	Gráficos

Figura 2.9: Síntesis de las características del hardware ([GHV⁺06]).

automatizar el proceso de visualización. En los tres casos se busca registrar o guardar el proceso de visualización para luego reutilizarlo como sugerencias en las próximas ejecuciones. Cuando el usuario vuelva a ejecutar el proceso de visualización, los pasos que realizó en ejecuciones pasadas son sugeridos en esta nueva ejecución.

El principal problema que tiene esta estrategia es que las sugerencias se realizan a partir de lo que el usuario hizo; si tenemos en cuenta que el usuario no siempre tiene los conocimientos necesarios para crear una visualización efectiva, no hay ninguna seguridad de que las sugerencias sean correctas; es decir, no hay barreras que eviten que el usuario introduzca decisiones erróneas.

Finalmente, en 2008, se presentó comercialmente el sistema *Tableau Software* el cual funciona como un sistema de visualización, análisis y consulta de bases de datos multidimensionales. Este sistema está basado en Polaris ([SH00], [STH02], [STH08]). Polaris es un lenguaje visual declarativo de consultas que permite especificar representaciones gráficas 2D, similar al diseño gráfico que definió Mackinlay más de 20 años antes. Este lenguaje tiene tres elementos base; un álgebra de tabla que permite capturar la estructura de las tablas de una base de datos y su codificación especial, una taxonomía gráfica para la especificación de tipos gráficos y finalmente un sistema efectivo para codificación visual.

Utilizando el álgebra de tabla, Polaris puede determinar las características de los tipos de datos a representar y en función de esto la mejor técnica de representación, utilizando un ranking como se ve en la Figura 2.2. La importancia de las relaciones y

los objetos a representar también es relevante en este sistema; un cambio en el orden de importancia muchas veces genera un cambio en la representación. La automatización que presenta el sistema reside en la elección de la técnica a utilizar; el usuario solamente tiene que indicar los elementos a representar y el sistema selecciona la técnica adecuada para hacerlo. Originalmente este sistema clasificaba los datos en nominal, ordinal, cuantitativo e intervalo pero luego éstos fueron reducidos a ordinal y cuantitativo solamente.

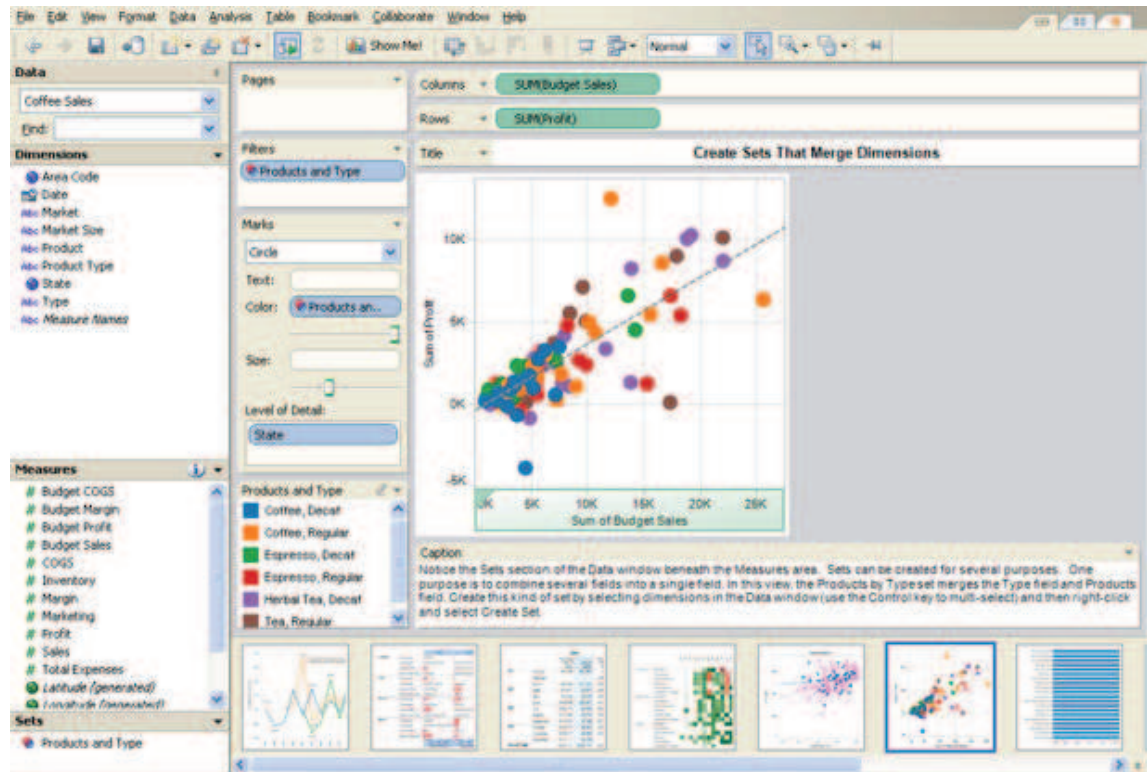


Figura 2.10: Captura de pantalla de Tableau Software.

2.2. Modelos de Visualización

Mientras que las nuevas tecnologías tanto en hardware como en software contribuyen a un aumento en el volumen y la diversidad de los datos que las organizaciones manejan, la exploración y la visualización de estos datos se tornan cada vez más dificultosas. Por esto, y debido al crecimiento en la cantidad y variedad de usuarios y a la demanda de mayor funcionalidad, es deseable que los sistemas de Visualización de Información (VI)

cumplan con propiedades como la extensibilidad, la modificabilidad, la predecibilidad y la posibilidad de estar integrados por componentes intercambiables.

En los últimos años han sido presentados numeros ambientes de VI; además, existen paquetes estándar que implementan la mayoría de las técnicas de visualización conocidas. A pesar de que estos sistemas no poseen todas las propiedades mencionadas, sus características similares permiten afirmar que el área ha madurado significativamente y hoy en día es posible desarrollar un modelo de referencia que satisfaga dichas propiedades. Tal modelo de referencia deberá permitir la especificación e integración de los componentes de hardware y software necesarios para la implementación de sistemas de VI para los diferentes dominios que han surgido recientemente, tales como las consultas a bases de datos, *data mining*, monitoreo de sistemas, etc.

En este contexto se presenta un modelo de referencia para los procesos de visualización. Si bien actualmente existen modelos aplicables a los distintos campos de la visualización, ninguno se consolida como un modelo unificado en todas las áreas de visualización.

2.2.1. Características del Proceso de Visualización

Las técnicas de visualización pueden ser utilizadas en una amplia gama de dominios de aplicación. De la gran variedad de dominios de aplicación de la visualización surgen precisamente los distintos campos de la disciplina. Si bien el objetivo común es la obtención de representaciones visuales interactivas con el propósito de ampliar la adquisición y el uso del conocimiento, según la naturaleza y características de la información a visualizar podemos hablar de:

- Visualización Científica: Tiene como característica la visualización de datos científicos, típicamente datos físicos. El sustrato espacial se encuentra presente en los datos a visualizar. Dentro de este campo a su vez se distinguen:
 - Visualización de Volúmenes: Representación, manipulación y rendering de datos volumétricos.
 - Visualización de Flujos: Representación, manipulación y rendering de datos vectoriales y tensoriales.
- Visualización de Software: Incluye la visualización de algoritmos y de programas (datos y código). Usualmente se distingue entre:

- Visualización de Algoritmos: Visualización de las estructuras de alto nivel que caracterizan a un software. Enfoque básicamente pedagógico.
- Visualización de Programas: Visualización del código y de los datos reales de un programa. Enfoque centrado en herramientas de la ingeniería de software.
- Visualización de Información: Visualización de datos abstractos, no basados en lo físico. Los datos no poseen un mapeo espacial inherente.

En una primera instancia, el diseño de un conjunto de interacciones bien definidas requiere de conocimiento en el dominio específico de la aplicación. Esto es natural ya que cada disciplina, y las características particulares de las mismas, determinan los análisis a los que se someterán los datos. Afortunadamente, aunque sea frecuente que distintos dominios de aplicación requieran representaciones visuales diferentes, varios de ellos pueden compartir estados intermedios de los datos, o requerir manipulaciones similares a nivel de vistas o incluso necesitar de las mismas transformaciones de datos.

Es claro que los requerimientos impuestos por cada dominio pueden llegar a ser dramáticamente diferentes; sin embargo, como se expresó previamente, de un análisis cuidadoso de los mismos podemos encontrar denominadores comunes:

- En los extremos de todos los procesos de visualización se encuentran los datos propios de cada dominio de aplicación, en uno, y una representación abstracta de esos datos en pantalla, en el otro. Las propiedades distintivas de los datos manejados en los extremos del proceso permite hablar de separación entre datos y vista (Figura 2.11).
- Esa separación entre datos y vista se hace más evidente en la visualización de información dado que los datos a ser mostrados carecen usualmente de un mapeo espacial inherente.
- Es evidente que entre los dos extremos mencionados los datos van sufriendo sucesivas transformaciones.
- Por otro lado, la visualización tiene que ser el medio por el cual el usuario explora el espacio de datos de su interés, para lograr obtener un *insight* de los mismos. Es claro que un proceso con esas características debe ser altamente interactivo.

- Para que los usuarios puedan interactuar de manera efectiva debe ser claro y natural sobre qué se está trabajando y cuáles son los posibles efectos de las interacciones propuestas.

Lo expuesto nos conduce a la necesidad de contar con un modelo consistente para los procesos de visualización que sea válido para los diferentes dominios de aplicación. Esa consistencia permitirá a los usuarios lograr interacciones efectivas en el proceso de visualizar datos provenientes de distintos dominios de aplicación, basándose en un único modelo mental. De esta manera, se facilitará la interacción con el sistema disminuyendo así el abismo de ejecución, i.e. *gulf of execution*.

La existencia de un modelo tal no sólo beneficia a los usuarios, sino también a los diseñadores. Al momento de extender un sistema existente para incluir nuevos dominios de aplicación, los diseñadores poseen un marco de referencia que les define cuáles son las transformaciones que deben sufrir los datos y cuál es el conjunto básico de operaciones que deberán proveer.

Lo que se busca básicamente es proveer un modelo que sea consistente con las posibles intenciones de los usuarios y brindar un marco de referencia en el que sea claro cómo interactúan las posibles transformaciones y las operaciones provistas. El modelo debe asistir tanto a los usuarios como a los diseñadores en la evaluación y ejecución de las tareas adecuadas para alcanzar las metas propuestas. En otras palabras se debe disminuir el abismo de evaluación, i.e. *gulf of evaluation*.

Una característica que debe proveer el modelo es permitir que los usuarios puedan enfocar su atención en las operaciones necesarias para lograr un determinado resultado, como así también en los operandos en los diferentes estadíos que abarca el proceso. Un ejemplo de modelo enfocado en las operaciones es el *pipeline* tradicional de visualización en el que el enfoque se realiza fuertemente en el proceso más que en el estado de los datos. En el otro extremo y como ejemplo de un modelo enfocado directamente en los operandos podemos nombrar a las planillas de cálculo, las cuales muestran los distintos estados de los datos manipulados más que las relaciones existentes entre ellos.

Si bien los investigadores en el área han presentado distintos modelos, generalmente éstos estaban enfocados a uno de los campos de la visualización en particular y no presentan las características propuestas. Ejemplos de este trabajo previo son el pipeline de W. Schroeder para visualización científica ([SML04]), el pipeline presentado por S. Card



Figura 2.11: Una vista de alto nivel al proceso de visualización.

([CM97]) para visualización de información y el modelo de estados de datos presentado por E. Chi ([CR98]). En ninguno de estos casos se logran conjuntamente los objetivos propuestos.

2.2.2. Modelo Unificado de Visualización

El Modelo Unificado de Visualización consiste en un único modelo que permite enfocarse tanto en los procesos como en los estados de los datos y que es aplicable a cualquier visualización independientemente del campo particular de origen. En este modelo quedan explícitamente representadas las operaciones provistas, los operandos sobre los que se pueden aplicar, conjuntamente con la secuencia de transformaciones propias del proceso. Este modelo constituye un marco conceptual en el que se pueden definir las interacciones necesarias, pudiéndose determinar sobre quién se opera, qué resultados se obtienen y cómo impacta sobre el proceso general.

Intuitivamente, cualquier proceso de visualización puede pensarse como una transformación de los datos en una representación visual; es un proceso cognitivo en el cual el usuario tiene que poder interactuar para lograr el objetivo buscado.

El Modelo Unificado de Visualización es un modelo de estados representado como un flujo entre los distintos estados que van asumiendo los datos a lo largo del proceso. En el esquema planteado, los nodos representan los estados y las aristas, las transformaciones necesarias para pasar de un estado al próximo. El modelo consiste en cinco estados y cuatro transformaciones (Figura 2.12), más las interacciones sobre cada estado y transformación.

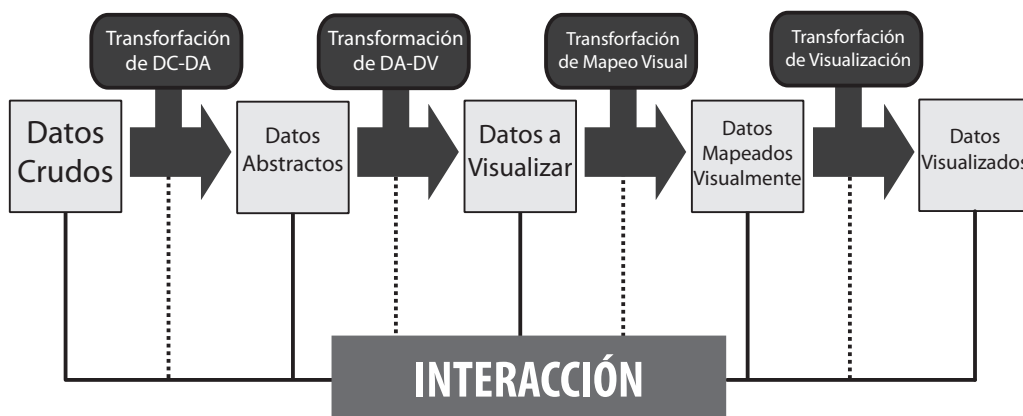


Figura 2.12: Representación gráfica del Modelo Unificado de Visualización.

Se puede describir entonces el modelo en términos de los estados y las transformaciones como se detalla a continuación:

- **Estado: Datos Crudos:** Son los datos provenientes del dominio de aplicación.
- **Transformación: Datos Crudos \mapsto Datos Abstractos:** Esta transformación permite al usuario seleccionar cuáles son los datos que quiere visualizar y llevarlos a un formato manejable por el sistema. Durante esta transformación se generan los metadatos.
- **Estado: Datos Abstractos:** Son los datos potencialmente visualizables por el usuario. Además de los datos propiamente dichos, también se dispone de los metadatos generados en la transformación anterior.
- **Transformación: Datos Abstractos \mapsto Datos a Visualizar:** Esta transformación permite que el usuario indique qué quiere visualizar dentro del conjunto de Datos Abstractos
- **Estado: Datos a Visualizar:** Son todos los datos que van a estar presentes en la visualización. Puede ser un subconjunto de los datos del estado anterior. Es importante notar que para un mismo conjunto de Datos Abstractos se pueden generar varios conjuntos de Datos a Visualizar.
- **Transformación: Transformación de Mapeo Visual:** Esta transformación permite que el usuario especifique cómo quiere visualizar los datos presentes en el

conjunto de Datos a Visualizar. Se generan las estructuras necesarias para soportar el sustrato espacial, los elementos visuales que se utilizarán en las representación y los atributos de los mismos.

- **Estado: Datos Mapeados Visualmente:** Son los datos a visualizar enriquecidos con información necesaria para su representación en pantalla. Dado que un mismo conjunto de datos puede ser mostrado de varias maneras distintas en pantalla, resulta natural el hecho de que pueda existir más de un conjunto de Datos Mapeados Visualmente para un conjunto de Datos a Visualizar dado.
- **Transformación: Transformación de Visualización:** Esta transformación permite la presentación en pantalla de todos los datos contenidos en el conjunto de Datos Mapeados Visualmente. Usualmente resultará en la aplicación de una técnica de visualización que soporte el mapeo visual presentado.
- **Estado: Datos Visualizados: Datos Visualizados:** Son los productos finales del proceso. En realidad, constituyen el punto de partida para el proceso de exploración que deberá realizar el usuario. Cada representación visual se corresponde con un conjunto de Datos Mapeados Visualmente.

Los estados de los datos

A lo largo de todo el proceso, los datos van sufriendo transformaciones desde su estado en el dominio de aplicación hasta llegar a obtener una representación visual de los mismos. La distinción entre estados mostrados no siempre obedece a diferencias estructurales sino también a los diferentes roles que cumplen dentro de las secuenciación de todo el proceso. En algunos casos, la distinción sólo tiene como justificativo la representación conceptual de un determinado estado, siempre teniendo como objetivo la consistencia con el modelo conceptual del usuario.

Como se puede observar en la Figura 2.12 se distinguen los siguientes estados:

- **Datos Crudos:** Se corresponde con el estado inicial. La clasificación de los datos es un tema muy amplio y complejo, y será abordado haciendo énfasis en las características que puedan resultar interesantes para su posterior visualización. Bertin ([Ber83]) sugiere dos formas básicas de datos: Valores de Datos y Estructuras

de Datos. En otras palabras clasifica los datos en entidades y relaciones. Estos conceptos tienen una larga historia en el diseño de Bases de Datos pero su uso en Visualización es reciente.

- **Entidades:** Son los objetos de interés que queremos visualizar.
- **Relaciones:** Las relaciones pueden estar definidas por las estructuras y/o patrones bajo los que se presentan las entidades. Las relaciones pueden estar en forma explícita dentro de los Datos Crudos u obtenerlas puede llegar a ser el objetivo del proceso de visualización. Las relaciones pueden ser físicas, conceptuales o estructurales. Tanto las entidades como las relaciones poseen propiedades que se expresan en términos de sus atributos.
- **Atributos:** Expresan las propiedades de las entidades y de las relaciones; no pueden ser pensados en forma independiente. Un aspecto importante a considerar al momento de diseñar una visualización son las características de los atributos a mostrar.

Cuando nos referimos a datos crudos consideramos que éstos pueden provenir no sólo de fuentes externas, sino también de visualizaciones previas.

- **Datos Abstractos:** Es un estado intermedio de los datos, en el que éstos están en un formato manejable pero aún no visualizable. En este estado tenemos todos los datos que el usuario preseleccionó dentro de los Datos Crudos como potencialmente visualizables. Dentro de este estado podemos tener entonces un subconjunto de los Datos Crudos al cual se le pueden haber agregado datos derivados de los existentes o también pueden ser Datos Crudos que se pueden haber mejorado de alguna manera. En síntesis, los Datos Abstractos constituyen el universo visualizable. Cualquier interacción del usuario que exceda a los datos almacenados en este estado demandará una nueva captura de Datos Crudos.
- **Datos a Visualizar:** Este estado presenta un subconjunto de los Datos Abstractos. Está constituido por todos los datos que van a ser visualizados de alguna manera. Es un estado que se justifica claramente desde lo conceptual. Es importante tener en cuenta que para un mismo conjunto de Datos Crudos se puede tener entonces más de un conjunto de Datos a Visualizar. Estos datos pueden no ser aún directamente visualizables.

- **Datos Mapeados Visualmente:** Éste es el estado previo a la generación de la representación visual. Es el resultado de mapear visualmente los Datos a Visualizar. Los datos presentes en este estado son directamente visualizables aplicando una técnica que los soporte. La característica distintiva de este estado es la presencia de un sustrato espacial que directamente pudo haber estado ausente en el estado anterior; precisamente, ésta es una de las diferencias básicas entre Visualización de Información y Científica. Podemos decir que: *Datos Mapeados Visualmente = Datos a Visualizar + Estructura Visual* donde la Estructura Visual está determinada por *Estructura Visual = Sustrato Espacial + Elementos Visuales + Atributos gráficos de los elementos visuales*. En este punto resulta evidente que para un determinado conjunto de Datos a Visualizar pueden coexistir varios conjuntos de Datos Mapeados Visualmente. Esta característica es relevante en el marco de un proceso de exploración del espacio de datos, donde el usuario puede o necesita comparar y analizar distintas maneras de visualizar un mismo conjunto de datos.
- **Datos Visualizados:** Es el último estado del proceso de visualización. Esto no significa que sea el estado final del proceso cognitivo que está llevando a cabo el usuario; todo lo contrario, estos constituyen su espacio de exploración. Es muy probable que a partir de una visualización el usuario comience a interactuar para lograr el *insight* de los datos que es el objetivo buscado.

Las transformaciones

Son los procesos que permiten pasar de un estado al próximo; en todos los casos son procesos que forman parte de una aplicación interactiva pudiendo por lo tanto requerir interacción del usuario. Como se desprende de la Figura 2.12, vemos que en el modelo se distinguen cuatro transformaciones:

- **Transformación de Datos Crudos a Datos Abstractos:** Esta transformación permite transformar los Datos Crudos en Datos Abstractos. Es decir, a partir de las fuentes de datos externas o provenientes de otras visualizaciones seleccionadas por el usuario deberá generar los Datos Abstractos. Esta transformación puede generar un nuevo conjunto de Datos Abstractos o directamente incorporarlos a un conjunto existente según lo indique el usuario.

- **Transformación de Datos Abstractos a Datos a Visualizar:** De todos los datos y relaciones seleccionados por el usuario para la generación de los Datos Abstractos es frecuente que éste solo quiera visualizar un subconjunto. Esta transformación permite realizar ese trabajo de selección, proyección y filtrado. En función de lo expresado por el usuario se generan nuevos conjuntos pertenecientes al estado de Datos a Visualizar, manteniendo para todos ellos el mismo conjunto de Datos Abstractos. El objetivo último de esta transformación es permitir que el usuario defina qué es lo que quiere visualizar en una determinada instancia, sin preocuparse todavía del cómo.
- **Transformación de Mapeo de Visualización:** Es una de las transformaciones clave en todo el proceso de visualización. Es una transformación netamente interactiva y permite definir al usuario cómo quiere visualizar sus datos. Es la transformación encargada de realizar el mapeo visual, es decir que se determinará:
 - Cuáles son las estructuras visuales adecuadas.
 - Qué atributos se mapearán espacialmente y cómo.
 - Qué elementos visuales se utilizarán y con qué atributos gráficos

El objetivo último de esta transformación es lograr un mapeo **expresivo** y **efectivo**. Decimos que un mapeo es **expresivo** cuando son representados todos y absolutamente todos los datos pertenecientes al conjunto de Datos a Visualizar. Por otro lado, la **efectividad** del mapeo estará dada por la manera en que la representación visual sea percibida por el usuario. Como puede intuirse, esta transformación no es trivial y gran parte de su éxito depende del conocimiento que se le imprima de las características perceptuales del humano. Esto lleva a concluir que el usuario debe explorar y comparar distintas alternativas de representación hasta lograr los resultados buscados.

- **Transformación de Visualización:** Es la transformación encargada de generar la representación visual en pantalla según lo expresado en los Datos Mapeados Visualmente. Para un determinado conjunto de Datos Mapeados Visualmente pueden existir varias técnicas que lo soporten, debiendo el usuario optar por alguna de ellas para obtener de esta manera los Datos Visualizados.

Las interacciones

Se ha expresado reiteradamente que la visualización debe ser un proceso interactivo. La visualización es un proceso cognitivo mediante el cual iterativa e interactivamente el usuario explora el espacio de sus datos. Se consideran interacciones a aquellas ofrecidas por la visualización, por el proceso de visualización o a las que ocurren a nivel cognitivo. Con respecto a las primeras dos, es claro que éstas puede ocurrir en distintos puntos a lo largo de todo el proceso de visualización. Una vez realizada la primera interacción, es decir una vez que se obtuvo la primera representación visual, el usuario interactúa sobre la última etapa del proceso. El hecho de que las interacciones tengan lugar en el estado de los Datos Visualizados no quiere decir que necesariamente se resuelvan en ese estado.

La interacción está reflejada en el modelo, sin duda, a nivel de los distintos estados y las correspondientes transformaciones. Las interacciones en el MUV fueron separadas en dos grupos, interacciones básicas y de alto nivel. Las interacciones básicas son aquellas que manipulan directamente los estados de los datos y las transformaciones del proceso. Son interacciones simples o elementales. Las interacciones de alto nivel, abarcan aquellas interacciones complejas características de un proceso de visualización.

Un ejemplo de interacción de alto nivel es *comparar*. Una persona puede ejecutar el proceso de visualización para obtener una visualización que le permita comparar fácil y visualmente un conjunto de datos. Las interacciones de alto y bajo nivel también pueden ser referidas como *tareas de alto* y *bajo nivel* respectivamente. Las de bajo nivel representan la forma de lograr aquéllas de mayor nivel de abstracción. Estas tareas son aplicables en cada estado de los datos y en las respectivas transformaciones. Por ejemplo, en el estado *Datos Crudos* se tiene la tarea *seleccionar fuente de datos*; esta operación es de bajo nivel y permite seleccionar la o las fuentes de datos que se considerarán datos crudos. En la transformación de *Datos Crudos* a *Datos Abstractos* también podemos tener tareas, por ejemplo *consultar estructura de datos crudos*; esta operación le permite al usuario consultar la estructura de los datos que se está manipulando.

Ejemplos

Para mostrar cómo el modelo soporta aplicaciones provenientes de las distintas ramas de visualización se tomaron distintos ejemplos representativos y se encuadraron en el modelo. Para cada aplicación, partiendo de los datos crudos, se aplican las

sucesivas transformaciones generándose así los sucesivos estados intermedios hasta alcanzar finalmente las representaciones visuales correspondientes.

En el ejemplo de Visualización Científica que se muestra (Figura 2.13), se pueden observar distintas visualizaciones posibles de acuerdo a las elecciones hechas en cada etapa del proceso. En este caso, se supone que se cuenta con un conjunto de tomografías y se elige una de éstas para ser visualizada. De acuerdo a distintas necesidades, se puede querer ver el conjunto 3D de una determinada tomografía o sólo parte de ésta. Cuando se trata de visualizar esta última opción se puede querer visualizar sólo un tipo de tejido (como el óseo en este ejemplo) o una parte específica del conjunto volumétrico.

En el ejemplo de Visualización de Información (Figura 2.14) el objetivo es visualizar distintos aspectos de información sobre llamadas telefónicas realizadas por los abonados a una empresa de telefonía. En estos ejemplos podemos ver cómo diferentes conjuntos de datos se visualizan de diferente forma y además existen diferentes formas de visualizar un mismo conjunto de datos.

2.3. Taxonomías

Todo concepto evoluciona a través de tres estados de formalidad ([DBDH05]), terminología, taxonomía y ontología, siendo este último el más formal. Es por esto que son tan importantes las taxonomías de visualización para este trabajo. Como nuestro objetivo es construir ontologías para dar soporte a la semántica y al proceso de razonamiento en el *pipeline* de visualización, es posible usar las taxonomías ya definidas y dar el paso siguiente en formalidad. Las taxonomías que analizaremos pueden ser agrupadas en 4 categorías: taxonomías de datos, de tareas, del proceso de visualización y otras

La primera clasificación presentada en el área de Visualización y Computación Gráfica fue la desarrollada por Jacques Bertin, “Semiology of Graphics”, ([Ber83]). Serge Bonin, discípulo de Bertin, explica que en el laboratorio de gráficos solían ocuparse de ayudar a los investigadores con los diagramas de sus publicaciones. Se percataron de casi nadie los miraba y aún eran menos los que los entendían. Ello les llevó a decidir entre simplemente redibujar con mano experta lo que se les pedía o tratar de analizar y entender el contenido del dibujo, reconstruir la tabla de datos y recrear el gráfico buscando la comprensión rápida de la información y la aparición de información visualmente relevante.

Al igual que otros investigadores han hecho posteriormente en el ámbito de las representaciones gráficas, la búsqueda del por qué de la inutilidad de los mismos los condujo a reflexionar sobre cómo realizarlos de forma que fueran útiles, a identificar las variables visuales y a encontrar las reglas de construcción de los mismos. La Semiología, propuesta por Ferdinand de Saussure, es la ciencia que estudia los signos que se utilizan en la comunicación. Así pues, Bertin y su grupo derivó en el estudio de los signos visuales y de su reglas “gramaticales”. La base del trabajo de Bertin es el reconocimiento de que “los gráficos son un conjunto de signos que permiten transcribir las relaciones de diferencia, de orden o de proporcionalidad existentes entre los datos cualitativos o cuantitativos”. Para Bertin, la representación gráfica cumple una doble función: como memoria artificial y como instrumento de descubrimiento, ligada a la enorme potencia de la percepción visual. Una característica particular de la clasificación de Bertin es que combina en una misma taxonomía elementos visuales, con las tareas y con los tipos de datos. Este trabajo sirvió como base para el desarrollo de nuevas taxonomías, enfocadas en áreas particulares tales como la clasificación de datos, de tareas y de representaciones visuales.

En 1996, en el trabajo “A Representational Analysis of Relational Information Displays” ([Zha96]) se presenta una taxonomía de *displays* de información relacional (*Relational Information Displays, RIDs*), basada en las características de los datos a representar. Las propiedades de los RIDs son determinadas por las propiedades de sus dimensiones y cada dimensión tiene una escala (ratio, interval, ordinal y nominal). Zhang define dos tipos de dimensiones, **representados** y **representaciones**.

- **Representado:** son las dimensiones de un dominio original que van a estar representados por las dimensiones físicas de la RID.
- **Representación:** son las dimensiones físicas de la RID que se utilizan para representar las dimensiones del dominio original.

La diferencia que se establece con estos dos tipos de dimensiones tiene su justificación en la separación entre las dimensiones que se representan y las dimensiones usadas para representarlas. Es decir, si se quiere representar un costo monetario, la dimensión **representada** es el costo y si la representación de este elemento se va a realizar a través de posiciones en un gráfico de coordenadas paralelas entonces la dimensión **representación** es la distancia en eje de coordenadas.

Al igual que Roth ([RM91]), Zhang establece un conjunto de acciones que se pueden ejecutar sobre una RID o como él lo presenta, una taxonomía de tareas sobre RIDs. Su taxonomía está compuesta por tres tareas:

- Recuperación de Información. (*Information Retrieval*)
- Comparación dentro de una misma dimensión o entre dimensiones. (*Comparison Within-dimensions or Between-dimensions*)
- Integración de Información (*Information Integration*)

Zhang justifica la creación de esta taxonomía debido a que la efectividad de una presentación no solo depende de la representación en sí sino también de la estructura de la tarea a realizar sobre la misma. Finalmente, en [Chi00], [QZP03] y [TM04] se presentan taxonomías de técnicas de representación en función de las características de los datos. Las tablas de taxonomías presentadas en dichos trabajos serán consideradas al momento de definir las ontologías para visualización.

En el capítulo 4 se describirán en detalle las clasificaciones y taxonomías presentadas para la caracterización de los datos, de las tareas, del proceso de visualización y de otros elementos; así también como nuestra propuesta para cada una de éstas.

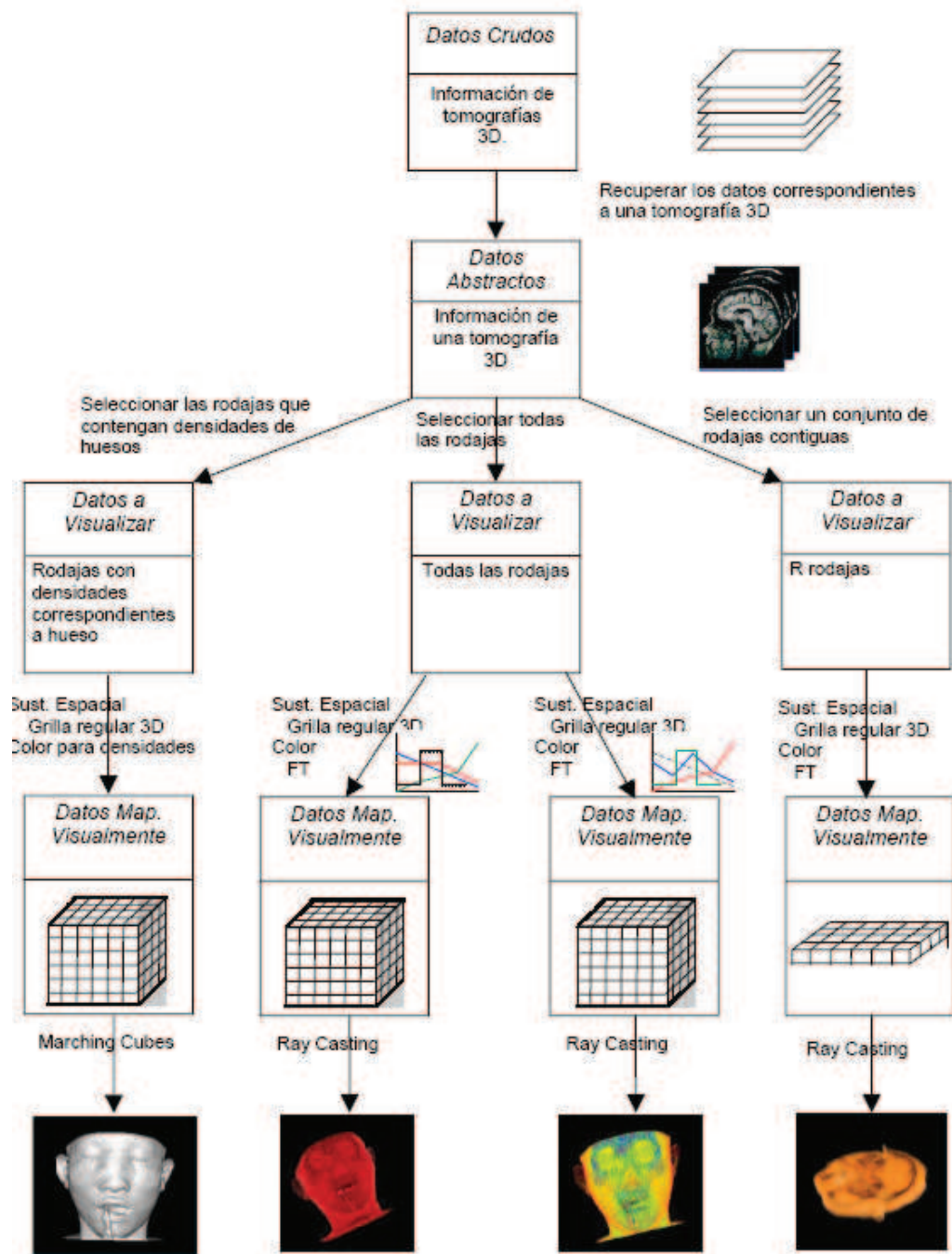


Figura 2.13: El Modelo Unificado de Visualización en Visualización Científica.

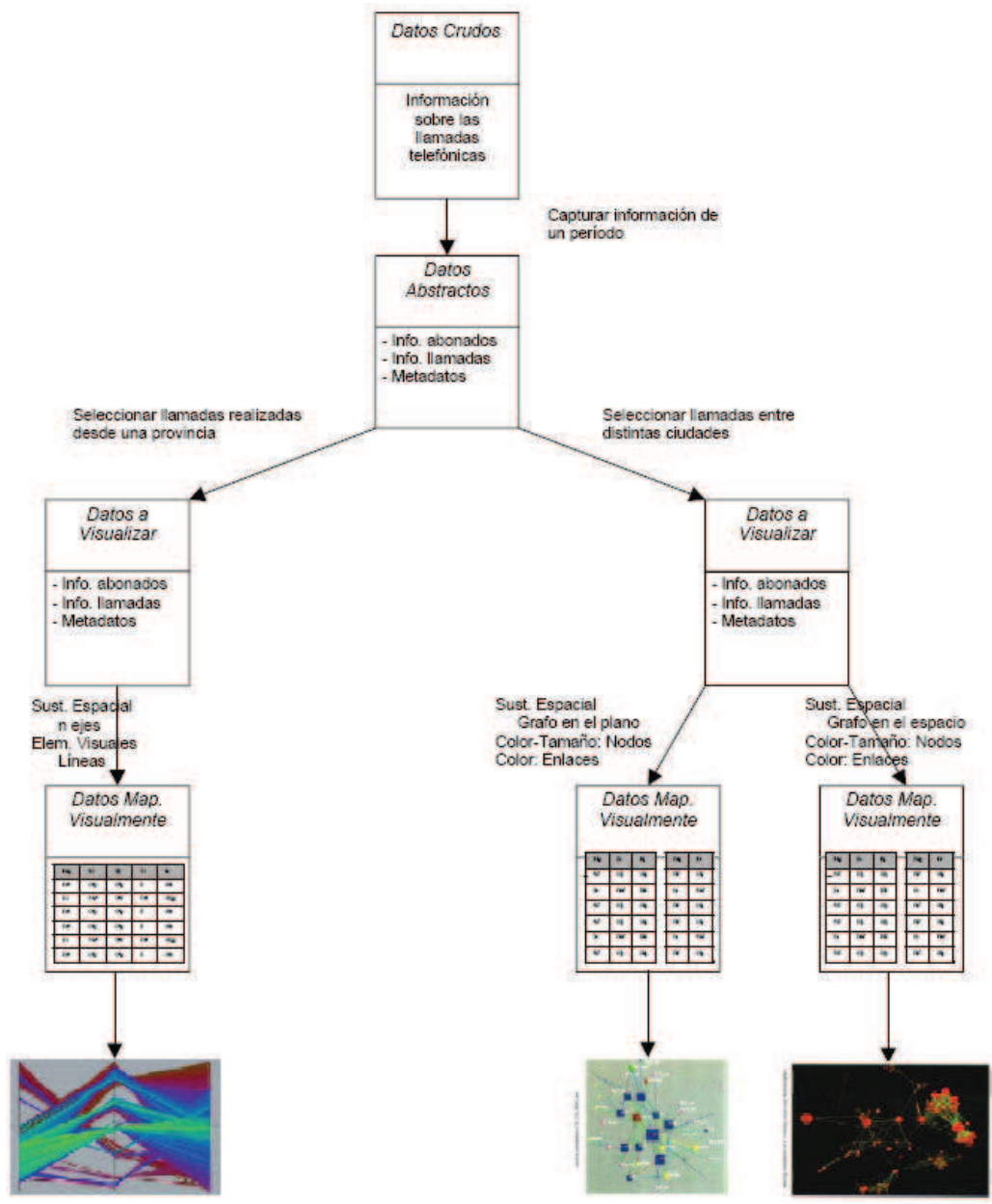


Figura 2.14: El Modelo Unificado de Visualización en Visualización de Información.

Capítulo 3

Visualización y Semántica

3.1. Visualización Basada en Semántica

Un proceso de visualización basado en semántica es aquél que utiliza el significado de los datos, del proceso de visualización y de los elementos involucrados en el mismo para asistir al usuario en la toma de decisiones y asegurar un resultado exitoso. El modelado de un proceso de visualización junto con sus interacciones no es suficiente para asegurar una visualización efectiva; una manera de lograrlo es desarrollar un meta modelo, basado en semántica, que incorpore la información necesaria para obtener una visualización exitosa. Un proceso de visualización basado en semántica debe ser independiente de la plataforma sobre la cual se ejecuta, fácilmente extensible y sobre todo estar basado en estándares. Para comprender cómo se constituye un proceso basado en semántica primero debemos comprender aquellos elementos propios de la semántica.

3.2. Semántica

La noción de información semántica se popularizó a través de la Web Semántica. La Web Semántica es una extensión de la World Wide Web en la que se definen la semántica de la información y los servicios existentes en la web. Actualmente la Web se enfrenta a un problema de interpretación de los contenidos. Una búsqueda sobre el término “La Pampa” incluirá resultados de la provincia de Argentina, la región geográfica y un método de laboratorio ([VHS07]); para un humano es fácil ver que estos tres elementos tienen

significados diferentes, gracias a sus conocimientos, pero para un sistema autónomo no. La Web Semántica busca solucionar éste problema. La Web Semántica es una capa que se le incorpora a la Web y permite que la misma pueda dar mejores resultados a los pedidos de contenido de las personas y de los sistemas. El objetivo de la Web Semántica es el de transformar la Web de un conjunto de repositorios de documentos enlazados a una base de conocimiento ([Hor07]). La Web Semántica se logra agregando anotaciones semánticas al contenido Web, es decir información que describe el significado de tales contenidos. Un buscador puede generar mejores resultados al considerar la información incluida en dichas anotaciones.

Sin embargo, en este punto el problema de la Web aún no está resuelto. El problema original de la Web es la incapacidad de los sistemas autónomos para interpretar el contenido presente; las anotaciones son más contenido por lo que el problema sigue siendo el mismo. Para dar una solución a este dilema aparecen las ontologías.

3.2.1. Ontología

El término “ontología” proviene de la filosofía y el direccionario de la Real Academia Española lo define como: “la parte de la metafísica que trata del ser en general y de sus propiedades trascendentales”. Es decir, lo que es, lo que existe, a lo que puede hacerse referencia y sus propiedades. En el contexto de Ciencias de la Computación se considera que una ontología es un modelo de aspectos del mundo o de cierto dominio particular. Una ontología introduce un vocabulario que describe los diversos aspectos del dominio que está siendo modelado y proporciona una especificación explícita del significado del vocabulario ([PRH⁺06], [FMK⁺08]). Para nuestros propósitos:

Una ontología es una especificación explícita y formal de una conceptualización compartida ([Gru93]).

La *conceptualización* se refiere a la definición de modelos abstractos para representar el mundo real. Dado que es una *especificación explícita* define de manera concreta los conceptos, relaciones entre ellos y restricciones sobre los conceptos y las relaciones que componen la ontología. La característica de *formal* se refiere a que la definiciones se expresan sin ambigüedad para poder ser interpretada por componente automáticos. Finalmente, la característica *compartida* refleja que el conocimiento capturado es consensuado por una comunidad de expertos en el dominio.

Las clases son el foco de la mayoría de las ontologías y permiten describir conceptos en un dominio dado. Por ejemplo, podemos considerar la clase **Forma** como el concepto que representa todas las formas posibles. Las formas específicas son instancias de esta clase. Una clase puede tener subclases que representan conceptos que son más específicas que la superclase. Por ejemplo, podríamos dividir a la clase **Forma** en **Círculo**, **Cuadrado** y **Triángulo equilátero**. Alternativamente, podemos dividir a la clase **Forma** en **Regulares** e **Irregulares**. Las propiedades de las clases y las instancias permiten describir sus características y las relaciones entre ellos: un triángulo en particular puede tener una base de tamaño *10cm* y una altura de *15cm*. Para representar esta información podemos incorporar al concepto **Triángulo equilátero** dos propiedades: la propiedad *base* y la propiedad *altura* con los valores 10 y *15cm*, respectivamente. Cuando una propiedad es de tipo *booleano*, *float*, *int*, *cadena*, *fecha*, *fecha y hora* o *tiempo* la llamamos “propiedad de dato” o “*data property*”. Si el valor de la propiedad es una instancia de una clase, entonces lo llamamos “propiedad de objeto” u “*object property*”. Las “propiedades de objeto” son útiles para establecer relaciones entre clases. Cada propiedad tiene asociado un dominio y un rango, los valores permitidos para las propiedades del objeto a menudo se llaman “el rango de la propiedad”. Las clases a las que una propiedad se adjunta se llama el dominio de la propiedad.

La arquitectura de la Web depende de normas acordadas y estandarizadas, tales como HTTP que permiten que la información sea compartida e intercambiada. Un lenguaje de ontologías estándar es, por tanto, una condición previa necesaria para que las ontologías se utilicen para compartir e intercambiar información. Reconociendo esta necesidad, la World Wide Web Consortium (W3C), creó un grupo de trabajo para desarrollar un lenguaje estandarizado. El resultado de este emprendimiento fue el lenguaje de ontología OWL ([PSHH04]). OWL se basa en los resultados obtenidos en otros lenguajes de ontologías tales como OIL ([FvHH⁺01]) y DAML+OIL ([MFHS02]) y, al igual que ellos, se basa en *Description Logics* (DL).

3.2.2. Description Logics

Las *description logics* son una familia de formalismos de representación del conocimiento basados en lógica; estos formalismos son descendientes de las *Semantic Networks* ([Woo85]) y KL-ONE ([SBI85]). Todos ellos utilizan un modelo orientado a objetos en el cual el mundo o dominio es descrito en términos de *individuos*, *conceptos*

(también llamados clases) y *roles* (también llamados relaciones o propiedades). Los *individuos* son los elementos básicos del dominio, e.g. “La Pampa”; los conceptos describen un conjunto de individuos que tienen características similares, e.g. “Provincia Argentina” y los roles indican relaciones entre pares de individuos, e.g. “tieneCapital”, “La Pampa tieneCapital Santa Rosa”. A través de DLs es posible establecer que un concepto es un subconcepto de otro o que son exactamente equivalentes. Por ejemplo, es posible decir, utilizando notación DL, $PadreFeliz \equiv Padre \sqcap \forall tieneHijo.(Inteligente \sqcup Deportista)$. Esta expresión introduce el concepto de *PadreFeliz* y establece que sus instancias son sólo aquellos individuos que son instancia de *Padre*, y todos sus hijos son instancias de *Inteligente* o *Deportista*. Otra característica distintiva de las DLs es que son lógicas, y por lo tanto tienen una semántica formal asociada. Las DLs puede considerarse como subconjuntos decidibles de predicados de lógica de predicados, con individuos que son equivalentes a las constantes, los conceptos a los predicados unarios y las funciones a los predicados binarios. Las DLs dan un significado preciso y sin ambigüedades a las descripciones del dominio; esto también permite el desarrollo de algoritmos de razonamiento que pueden ser utilizados para responder a preguntas complejas sobre el dominio. Dentro de la terminología propia de DLs existen tres términos que son mencionados regularmente: *ABox*, *TBox* y *KB*; *ABox* es la abreviatura de *Assertional Box* y representa la componente que contiene afirmaciones acerca de los individuos, *TBox* es la abreviatura de *Terminological Box* y representa la componente que contiene axiomas acerca de las clases y *KB* es la abreviatura de *Knowledge Base* y representa la combinación de una *ABox* y una *TBox*.

3.2.3. Ontology Web Language (OWL)

OWL significa “*Web Ontology Language*” y es un lenguaje para procesar información de la web. OWL se ubica como una extensión de RDF y está escrito en XML ([CFLGP03], [BvHH⁺04]). Mediante el uso de XML, un lenguaje en texto plano, la información de OWL se puede cambiar fácilmente entre diferentes tipos de sistemas que utilizan diferentes tipos de sistemas operativos y lenguajes de aplicación. Una ontología OWL contiene una secuencia de anotaciones, axiomas y hechos. Las anotaciones puede ser utilizadas para registrar la autoría y otra información relacionada con la ontología. El contenido principal de una ontología OWL se encuentra en sus axiomas y hechos, que proporcionan información sobre las clases, propiedades e individuos en la ontología.

OWL nació como el lenguaje de ontologías estándar para la web semántica. Es compatible con los lenguajes de ontologías anteriores, tales como OIL y DAML+OIL, y proporciona más poder para expresar semántica. Incluye la conjunción, disyunción, *existentially* y variables cuantificadas universalmente. Sin embargo, su poder expresivo presenta inconvenientes; algunos constructores del lenguaje son muy complejos, el proceso de razonamiento no es eficiente, no es fácil de usar y no es intuitivo.

OWL se presenta en tres versiones: *Full*, *DL* y *Lite*. OWL *Full* y OWL *DL* admiten el mismo conjunto de constructores. La diferencia radica en las restricciones sobre el uso de éstos y en el uso de las características de RDF. OWL *Lite* es un sublenguaje de OWL *DL* que sólo admite un subconjunto de los constructores de OWL. La razón principal de la existencia de OWL *DL* es que se han desarrollado potentes sistemas de razonamiento que se basan en ontologías limitadas por las restricciones necesarias para OWL *DL*.

3.2.4. Razonador Semántico

Un razonador semántico es una pieza de software que permite inferir consecuencias lógicas a partir de una base de conocimiento (datos + ontologías) ([MCGS08]). Supongamos que tenemos una ontología que contiene la descripción de *PadreFeliz* que se dio en la sección 3.2.2 y además sabemos que: *Miguel* tiene un hijo que se llama *Martín*, i.e. *Miguel tieneHijo Martín*, y *Martín* no es *Deportista*, sería provechoso poder inferir que *Martín* es *Inteligente*. Este ejemplo, que resulta algo trivial, muestra la capacidad de un razonador semántico y su utilidad ante consultas mucho más complejas. Una de las razones para usar DL como base para OWL fue la existencia de razonadores que operaban sobre DLs. El proceso de razonamiento es fundamental tanto para el diseño de ontologías de calidad, y la puesta en funcionamiento de ontologías en aplicaciones.

El razonamiento al momento del diseño de ontologías permite realizar una serie de tareas fundamentales para obtener un buen diseño. Estas tareas son: chequear si la definición de una clase es consistente consigo misma y con el resto de las definiciones de clases, chequear si la definición de una clase es más general que otra y finalmente construir una jerarquía de clases basada en sus definiciones. Este tipo de razonamiento se denomina *terminological reasoning* ([DFvH03]). Para el caso de la puesta en funcionamiento de ontologías en aplicaciones, el razonador permite realizar las siguientes tareas: encontrar la clase más específica que describe una instancia parcialmente especificada, encontrar todas

las instancias en un conjunto de datos que son instancias de una definición de clase dada, resolver consultas complejas y finalmente chequear la consistencia de una instancia con respecto a la ontología. Este tipo de razonamiento se denomina *instance reasoning*.

El proceso de razonamiento puede estar embebido en el mismo sistema que contiene las ontologías y los datos. La ventaja de este enfoque es que la comunicación entre el razonador y las ontologías+datos es directa; la desventaja es que los razonadores que pueden ser embebidos dentro de un sistema suelen no ser los apropiados para resolver consultas complejas. Este último caso puede ser resuelto mediante un razonador externo. Algunos razonadores disponibles en la actualidad son: Fact++ ([TH06]), Racer ([HM03]) y Pellet ([SPG⁺07a]). Utilizando un razonador externo se pueden resolver consultas más complejas pero hay un problema de eficiencia cuando las ontologías y/o los datos son grandes. Cuando se utiliza un razonador externo la comunicación entre el razonador y la ontología+datos se puede resolver de dos formas diferentes: una posibilidad es sólo enviar al razonador las partes de las ontologías y las instancias que son relevantes a la consulta que se quiere resolver. Esta solución minimiza el intercambio de información pero implica resolver cuáles son dichas partes, un proceso que a su vez requiere de razonamiento; otra opción es mantener una copia de la ontología y/o los datos en el razonador externo. El problema de esta opción es que para mantener ambas copias actualizadas se necesita intercambiar un volumen significativo de información.

3.3. Antecedentes

El estado del arte en esta área muestra buenos ejemplos de cómo la información semántica está siendo integrada al área de Visualización ([XCW05], [KCM06], [NHN02] and [WB98]). Sin embargo, en todos estos casos el rol de la semántica es el de mejorar la integración, consulta y descripción de la visualización; en ninguno de estos trabajos la información semántica es utilizada para crear la visualización. En [BGKS04] se puede encontrar una primera aproximación al objetivo planteado en esta tesis y en [RBG07] se observa un trabajo más cercano a nuestro objetivo.

El trabajo de Esteban Zimányi y Sabri Skhiri dit Gabouje ([BGKS04]) presenta un sistema en el cual se puede modificar una representación gráfica a partir de una descripción semántica. En este trabajo se define el concepto de *Modelo de Representación Adaptable*, que permite a los biólogos cambiar la semántica gráfica asociada a la semántica de los

datos. El enfoque también es genérico, ya que la semántica gráfica es común a varios tipos de redes bioquímicas. Los *Modelos de Representación Adaptables* son implementados utilizando XML; estos modelos se basan en un *XML Schema* que determina la correctitud del modelo. El modelo permite, actualmente, la definición de dos tipos de elementos: *Element* y *Link*. Estos se corresponden a los nodos y arcos de un grafo bioquímico. Un *Element* define la semántica gráfica asociada a un compuesto bioquímico (Figura 3.1), mientras que en *Link* define la semántica gráfica asociada al arco que conecta dos compuestos. Lamentablemente este trabajo sólo se realizó para utilizar los recursos biológicos; no aprovecha la capacidad de representación de RDF u OWL y no incluye ningún proceso de razonamiento a partir de la información semántica.

```

<Element>
  <Type>Compound</Type>
  <Glyph>
    <Shape>
      <Circle>
        <Radius>15</Radius>
      </Circle>
    </Shape>
    <BorderColor>None</BorderColor>
    <FillColor>None</FillColor>
    <Text>
      <Present>>true</Present>
      <Content>?ID</Content>
      <Position>Out</Position>
    </Text>
    <Icon>
      <Present>>true</Present>
      <Location>/image/compound.bmp</Location>
    </Icon>
    <BorderVisibility>>true</BorderVisibility>
  </Glyph>
</Element>

```

Figura 3.1: Ejemplo de cómo se describe la semántica gráfica. El compuesto se representa con un círculo de radio 15 en el cual se muestra el icono *compound.bmp*, la identificación del compuesto se muestra fuera del círculo ([BGKS04]).

Zhao Xu et. al introducen un *framework* para la integración de información semántica con información geográfica ([XCW05]), dicho *framework* se denominó GeoShare. Ese

sistema incluye un módulo de visualización el cual crea en forma automática mapas en formato *Scalable Vector Graphics* (SVG) embebidos con información semántica. El módulo gráfico, o *SVG Generation Service* (SVGGS), descrito en el trabajo crea una representación gráfica de un conjunto de datos geospaciales y luego le incorpora la información semántica asociada a esos datos. Aquí la información semántica se utiliza para enriquecer la representación gráfica pero no es usada en el proceso de creación del archivo SVG.

Kalogerakis et. al. presentan un *framework* ([KCM06]) que busca integrar el contenido gráfico de las escenas con el conocimiento específico del dominio para que los usuarios pueden consultar, interpretar, personalizar y manipular la información visualizada. Este trabajo se centró, particularmente, en la integración de escenas de realidad virtual con información semántica. Aunque este trabajo incluye la semántica en el proceso de creación de una escena virtual, su uso se limita a facilitar las acciones del usuario, no a guiarlas.

Nguyen et. al. ([NHN02]) y Weng Zu et. al. ([WB98]) también son ejemplos de desarrollos realizados sobre el área de Visualización y representaciones gráficas en donde se buscó integrar la información semántica. En estos casos, al igual que en los anteriores, el beneficio buscado fue el de mejorar los procesos de consulta y búsqueda pero no el de creación de las representaciones gráficas en sí.

Las técnicas de representación de volúmenes mapean atributos volumétricos, e.g. densidad, a estilos visuales. Comúnmente este mapeo se especifica mediante una función transferencia. La especificación de funciones transferencia es una tarea compleja y requiere conocimiento experto sobre la técnica de representación subyacente. En el caso de múltiples atributos volumétricos y múltiples estilos visuales, la especificación de una función transferencia multi-dimensional se vuelve más difícil y no es intuitiva. Rautek et. al. ([RBG07]) proponen una solución a este problema utilizando información semántica. En este trabajo se introduce el término *semantic layers* que permiten a un experto del dominio de aplicación especificar la función transferencia en un lenguaje natural. Una *semantic layer* define el mapeo de un atributo a un estilo visual. Los atributos volumétricos y los estilos visuales se representan como *fuzzy sets*. El mapeo se especifica a través de reglas que son evaluadas con aritméticas de *fuzzy logic* ([YZ92]). El usuario especifica los *fuzzy sets* y las reglas sin un conocimiento especial sobre la técnica de representación subyacente.

La componente central de este sistema son las reglas que permiten una descripción lingüística del mapeo deseado. Cada regla establece una premisa en la parte antecedente

y una conclusión en la parte consecuente. La consecuencia es una combinación lógica de valores semánticos con atributos de volumen (Figura 3.2). Este trabajo se limita solamente al *rendering* ilustrativo de volúmenes, no utiliza un método de representación estandarizado como OWL y no se incluye un proceso de razonamiento. Además, la semántica solamente se aplica a la función de transferencia, dejando por fuera otros elementos fundamentales en el proceso de visualización.

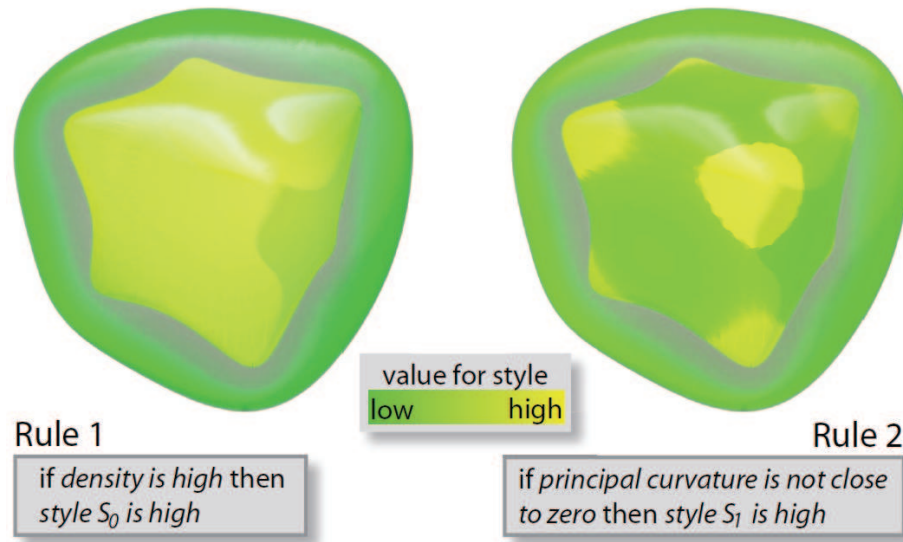


Figura 3.2: Ejemplo de *Semantic Layers* ([RBG07]). Un código de colores se utiliza para codificar los valores numéricos de cada estilo. El estilo de la izquierda muestra el resultado de una regla basada en la densidad. La imagen de la derecha muestra el resultado de una regla basada en la curvatura.

3.4. Semántica en el Proceso de Visualización

Nuestro proceso de visualización es el MUV, y como se describió en el capítulo anterior está constituido por estados y transformaciones. Sin embargo, el proceso como tal, no se encuentra aislado sino en un contexto (Figura 3.3). Es decir, los **estados** y las **transformaciones** son ejecutados por un **humano** para realizar una o más **tareas**, quien a través de **interacciones** actúa sobre el **proceso**. El proceso de visualización busca representar cierto conjunto de **datos** en un **contexto** dado. Estos elementos, el **proceso**, los **estados**, las **transformaciones**, el **humano**, las **interacciones**, los **datos**

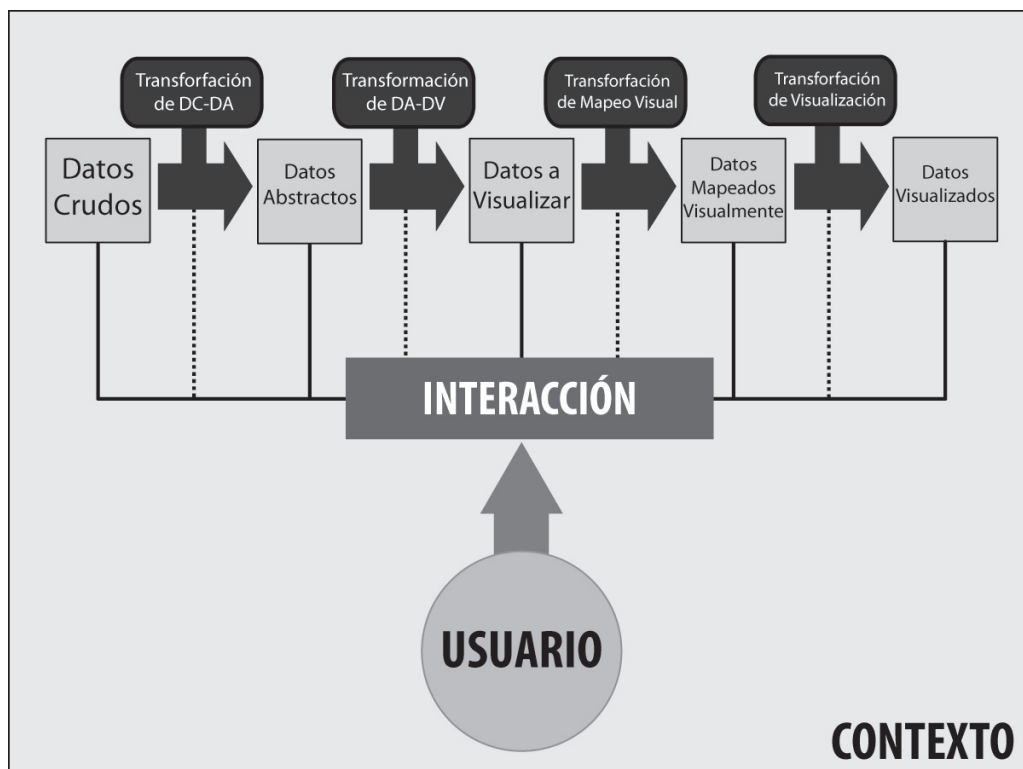


Figura 3.3: Elementos involucrados en el proceso de visualización: el proceso, los estados, las transformaciones, el usuario, las interacciones, los datos y el contexto.

y el **contexto** son los elementos que se deben caracterizar para poder incorporar semántica al proceso de visualización (Figura 3.4).

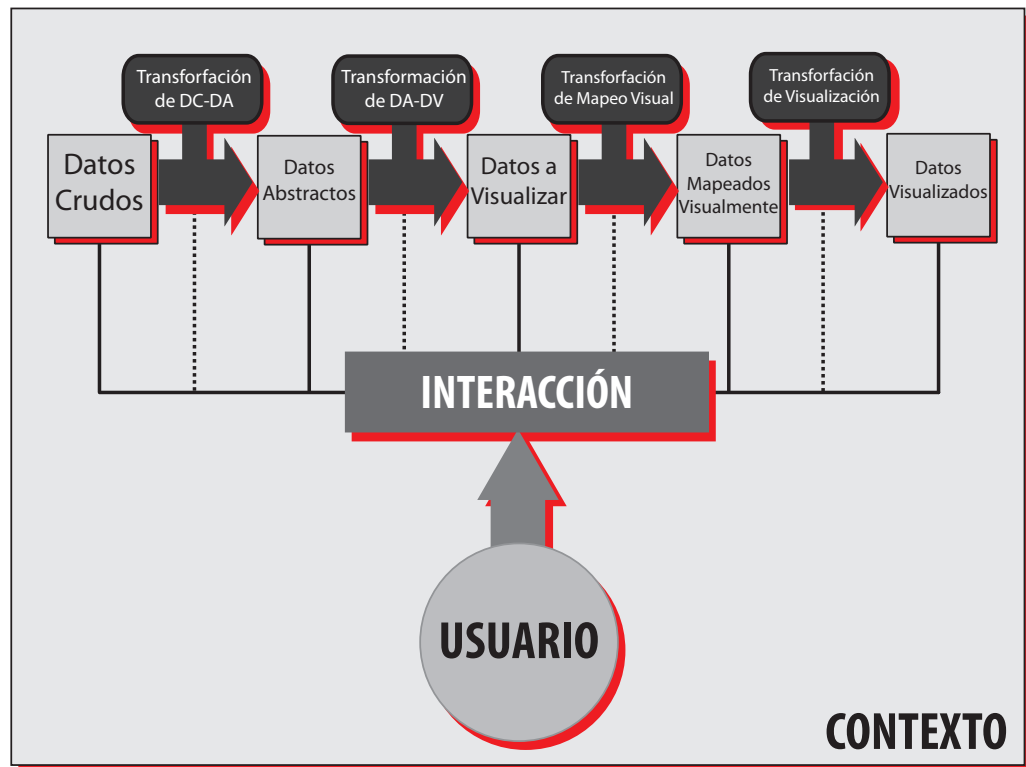


Figura 3.4: Cada elemento involucrado en el proceso de visualización adquiere semántica, representado aquí con el color rojo. Semántica involucrada en el proceso de visualización: semántica del proceso, de los estados, de las transformaciones, del usuario, de las interacciones, de los datos y del contexto.

Capítulo 4

Caracterizaciones

Existen tres niveles de formalismo: terminología, taxonomía o caracterización y ontología ([DBDH05]). Para obtener una visualización basada en semántica es necesario contar con ontologías que den soporte a dicha semántica. Éstas se pueden obtener a partir de las caracterizaciones existentes de los elementos involucrados en el proceso de visualización. Tales elementos son: los datos de entrada, las tareas a realizar, el humano, el contexto y finalmente el proceso de visualización. En este capítulo se presenta un análisis y una evaluación de las caracterizaciones existentes. Asimismo, en base al análisis y a las evaluaciones llevadas a cabo y teniendo en cuenta el trabajo previo se proponen nuevas caracterizaciones para cada uno de estos elementos en el contexto del proceso de visualización.

4.1. Caracterización de los datos en Visualización

Uno de los primeros pasos para automatizar el proceso de visualización, es formalizar la definición de los tipos de datos desde el punto de vista de la visualización. A continuación se presenta una revisión de las clasificaciones de datos que se tuvieron en cuenta para nuestro trabajo dada su relevancia en el proceso de visualización. Finalizaremos esta sección con la presentación de nuestra caracterización de datos que, además de cubrir los aspectos contemplados parcialmente en los antecedentes, los incluye y los integra a los que también consideramos deben incluirse en una caracterización adecuada para la visualización.

4.1.1. Antecedentes en caracterizaciones de datos

La primera clasificación que presentaremos es la primera clasificación formal presentada sobre las escalas de medidas. Introducida en 1946 por S. Stevens ([Ste46]), esta escala ha influenciado muchos trabajos posteriores. Stevens presenta cuatro tipos de datos, analizando para cada uno el conjunto de operaciones que se pueden aplicar sobre los mismos, la estructura matemática y los cálculos estadísticos posibles. Dicha información se ilustra en la Figura 4.1.

Escala	Operaciones básicas	Estructura matemática	Operaciones estadísticas
NOMINAL	Determinación de igualdad	<i>Permutation group.</i> $y = f(x)$, donde $f(x)$ es cualquier sustitución uno a uno.	Número de casos Modo Correlación
ORDINAL	Determinación de menor o mayor	<i>Isotonic group.</i> $y = f(x)$, donde $f(x)$ es cualquier función monotonica creciente.	Media Percentiles
INTERVALO	Determinación de igualdad entre intervalos o diferencia	<i>General linear group.</i> $y = ax + b$	Media Desviación estándar Correlación <i>rank-order</i> Correlación <i>product-moment</i>
RATIO	Determinación de igualdad de ratios	<i>Similarity group.</i> $y = ax$	Coefficiente de variación

Figura 4.1: Tipos de datos descriptos por S. Stevens en 1946. Para cada tipo se incluyen sus operaciones básicas, la estructura matemática que les da soporte y el conjunto de operaciones estadísticas que se pueden aplicar sobre los mismos ([Ste46]).

En 1984 se encuentra la primera mención ([CM84]), en forma indirecta, a una clasificación de datos en el contexto del proceso de visualización. Cleveland y McGill son los autores de dicho trabajo. Este artículo se centra en la definición de tareas perceptuales y en la evaluación de la exactitud con la cual un usuario puede extraer información de una visualización. Aunque este trabajo no cuenta con una sección dedicada a los tipos de datos, Cleveland y McGill hacen mención a los tipos de datos involucrados en sus representaciones gráficas.

En primer lugar, los autores mencionan que el principal objetivo del trabajo es la identificación de tareas perceptuales que son realizadas al extraer información cuantitativa de representaciones visuales. Todos los datos utilizados en el trabajo corresponden al tipo

de dato **cuantitativo**, es decir a valores que existen dentro de un rango de magnitudes. La falta de la definición de un tipo de dato no cuantificable, por ejemplo un tipo de dato cualitativo, se debe probablemente a que los autores tratan con aplicaciones correspondientes al área de Visualización Científica.

Dos años después, Jock Mackinlay ([Mac86]) continuó el trabajo presentado por Cleveland y McGill ([CM84]). Este es el primer trabajo en el que se define un proceso automático de visualización. Aunque Mackinlay no enfoca su trabajo en la clasificación de datos, sí establece que el tipo de un dato afecta la representación visual. Limitándose a representaciones gráficas en 2D, Mackinlay identifica tres tipos de datos:

- **Cuantitativo.** Todos aquellos datos que son rangos numéricos.
- **Ordinal.** Todo aquellos conjuntos de datos que pueden ser ordenados.
- **Nominal.** Todo aquellos conjuntos de datos que no pueden ser ordenados.

A diferencia del trabajo en el que se basa Mackinlay ([CM84]), aquí se consideran los elementos no numéricos, ordinales y nominales, como parte del conjunto de datos que pueden ser representados. Aunque la incorporación de los dos nuevos tipos de datos permiten extender significativamente el rango de datos a representar ninguno de estos trabajos caracteriza las relaciones que existen entre los datos.

Esta última caracterización comienza a tener relevancia a partir del trabajo de Steven Roth y Joe Mattis, ([RM90]). En este artículo se describe la necesidad de contar con una caracterización de los datos, la cual es una descripción de la semántica y la estructura de los datos a visualizar. La motivación para su desarrollo surge de preguntarse qué tipo de información sobre los datos a visualizar se debe proporcionar para lograr una visualización efectiva. Sin introducir el concepto de semántica queda establecida la importancia de la información sobre la información.

Para Roth y Mattis el conjunto de características, o dimensiones como ellos denominan, que son relevantes para la construcción de una representación visual son: **tipo de dato**, **propiedades de las relaciones**, **relaciones sobre relaciones**, **aridad** y **tareas del usuario**. Esta última característica, **tareas del usuario**, no está relacionada con la clasificación de los datos que estamos desarrollando, por lo que será considerada más adelante en este capítulo.

En la descripción del **tipo de dato** se diferencian tres características:

1. *Ordenamiento*. La naturaleza del ordenamiento de los datos es fundamental para la representación. De acuerdo a esto los datos pueden ser de uno de los siguientes tipos, cuantitativos, ordinales y nominales. La característica de ordenamiento presentada por Roth y Mattis se corresponde con la clasificación de los tipos de datos presentada por Mackinlay.
2. *Coordenadas vs. Cantidades*. Se debe diferenciar si los datos son elementos de un conjunto de coordenadas, es decir que cada elemento establece un punto o ubicación temporal, espacial u otra, o si los datos que representan cantidades no están embebidos en un marco de referencia.
3. *Dominio de pertenencia*. Roth y Mattis reconocen la necesidad de distinguir ciertos dominios de datos debido a que requieren un manejo especial en la representación. Como dominios de pertenencia consideran los conceptos de tiempo, espacio, temperatura y masa. Los datos temporales, por ejemplo, se mostrarán siempre sobre un eje horizontal y los datos que representen temperaturas, serán mapeados a un eje vertical.

Una segunda dimensión presentada en este trabajo corresponde a las **propiedades de las relaciones**. Esta característica describe la forma en que cada relación mapea elementos de un conjunto a otro. Esta forma de mapeo da lugar a tres propiedades:

1. *Cubrimiento relacional*. Esta característica indica si, a través de una relación específica, cada elemento de un conjunto de datos puede ser mapeado a por lo menos un elemento de otro conjunto de datos. Por ejemplo, si tenemos una relación *inicio-de-actividad* que mapea de un dato *actividad* a un dato *fecha* y además sabemos que la relación *inicio-de-actividad* tiene cubrimiento relacional entonces podemos estar seguros que toda actividad tendrá una fecha asociada. Se distinguen tres casos donde no habrá cubrimiento relacional:
 - Faltan datos. Por ejemplo, pueden existir actividades a las cuales les falten sus fechas.
 - La relación no es aplicable. Por ejemplo, una relación llamada *costo-de-transmisión-automática* que mapea de modelos de autos a costos no será aplicable para el caso de autos que no tengan transmisiones automáticas.

- El que no haya valores es informativo.
2. *Cardinalidad*. Esta característica indica el número de elementos de un conjunto al cual una relación puede mapear un dato de otro conjunto. Es decir, expresa la cantidad de valores que pueden ocurrir para un dato a través de una relación. Nuevamente se consideran tres posibilidades: Valor único, valor múltiple fijo y valor variable.
 3. *Unicidad*. Esta característica indica si para cada elemento de un conjunto, una relación mapea a un valor único.

La tercera dimensión presentada corresponde a las **relaciones entre relaciones**. No todas las relaciones son binarias, por ejemplo una relación *período-de-trabajo* relaciona un dato *empleado* con dos datos *año*, uno el año de inicio y otro el año de finalización. Roth y Mattis denominan a este tipo de dato, que contempla relaciones con aridad mayor a 2, tipo de dato complejo. Dentro de esta dimensión, junto a los tipos de datos complejos, también se encuentra el tipo *dependencia algebraica*. Este tipo de dato representa aquellos datos cuyo valor depende de una operación algebraica sobre otros datos. Este es el caso, por ejemplo, si se tuviese los datos *costo-material*, *costo-laboral* y *costo-total* y el dato *costo-total* fuera igual a *costo-material* más *costo-laboral*. El dato *costo-total* se puede ver como una agregación de los otros dos. Esta semántica del dato es muy significativa al momento de decidir la forma de representarlo.

Finalmente, exceptuando las **tareas del usuario**, la dimensión que se presenta es la de **aridad**. Para poder seleccionar técnicas apropiadas de representación es necesario conocer la **aridad** de cada relación. Esta última dimensión captura tal atributo.

El trabajo presentado por Roth y Mattis constituye la primera caracterización detallada de los datos a representar. Aunque los trabajos de Mackinlay & Cleveland y McGill también presentan una forma de clasificar los datos, no hay dudas de que el de Roth y Mattis es el primero en lograr una clasificación significativa ya que no sólo captura los posibles tipos de datos sino que además incorpora la semántica de los mismos y sus relaciones.

Dos años después de la publicación del trabajo de Mackinlay ([Mac86]), Stephen Wehrend y Clayton Lewis publican un trabajo en donde proponen un esquema de clasificación para los problemas de visualización y sus soluciones ([WL90])

independientemente del dominio de aplicación. Dicho esquema se construye a través de una matriz de objetos y operaciones. Los problemas se clasifican en función de la naturaleza de los objetos a representar, por ejemplo escalares o posiciones. Las categorías de objetos que se consideran incluyen: **dato escalar**, **campo escalar**, **nominal**, **dato direccional**, **campo direccional**, **forma**, **posición**, **región extendida en el espacio** y **estructura**.

La diferencia que plantean Wehrend y Lewis entre dato y campo, tanto para escalar como para direccional, está relacionada con la agrupación de los datos. **Dato** representa un solo elemento mientras que **campo** corresponde a una colección de datos.

Un año después de publicar su primer trabajo de clasificación ([RM90]), Roth y Mattis presentan una modificación a dicha clasificación ([RM91]). En ésta también utilizan un sistema de clasificación de datos, y extienden la descripción de las tareas del usuario dadas en el trabajo anterior. A diferencia del primer trabajo, en éste no se incluye la descripción de la aridad de las relaciones, aunque no se justifica su exclusión. Adicionalmente, cambian la nomenclatura que se utiliza para caracterizar los datos:

1. En lugar de dimensiones, se habla de **categorías**. Los datos no se caracterizan en una serie de dimensiones sino en una serie de categorías.
2. En lugar de hablar de los datos como elementos, i.e. los elementos de un conjunto de datos, ahora se habla de **objetos de base** o solo **objetos**.
3. En el primer trabajo las dimensiones eran: **Tipo de dato**, **propiedades de las relaciones** y **relaciones sobre relaciones**. Ahora, las nueva identificaciones para estas categorías son: **Características de los objetos de datos**, **características de las relaciones entre objetos de datos** y finalmente **relaciones entre relaciones**. La descripción de cada categoría se mantiene igual a la de su correspondiente dimensión, tal como se presentó en [Mac86].

En 1992 Brodly et. al. analizan el estado del arte de la visualización científica y presentan un marco de trabajo para la misma ([BCE⁺92]). Dentro de dicha presentación se introduce una clasificación de los datos, específica para la visualización científica. Consideran que los tipos de datos pueden ser: **puntos**, **escalares**, **vectores** o **tensores**. A diferencia de los trabajos presentados anteriormente, éste el es primero que introduce una notación formal para la descripción de los datos. Esta notación se denomina **notación E** y se describe a continuación.

Una entidad escalar de dimensión n , por ejemplo, se representa con la forma general E_n^S donde S identifica la entidad como escalar. Se utiliza V para vectores, T para tensores y P para puntos. La dimensión del vector se puede incluir como un sufijo, por ejemplo V_3 . De acuerdo a esta especificación, una clasificación de la forma $E_n^{V_5}$ representa un vector de 5 elementos definido sobre un dominio de n dimensiones.

El uso de un indicador de dimensión también genera una subclasificación de los tipos; en particular se consideran tres casos:

1. La entidad se define sobre un dominio continuo; en tal caso se utiliza la notación de dominio como se vio previamente.
2. La entidad está definida sobre regiones de un dominio continuo. Para esto se utiliza la notación $E_{|2|}$.
3. La entidad está definida sobre un conjunto enumerado. Esta característica se identifica con la notación $E_{\{1\}}$.

Es clara la influencia de la visualización científica en esta clasificación; en ningún momento se considera la posibilidad de un dato cualitativo como sí se mostró en los trabajos anteriores. Sin embargo, este trabajo es uno de los pocos que logra dar una descripción formal de la clasificación de los datos.

En 1994 Roth y Mattis presentan junto a John Kolojejchick y Jade Goldstein un nuevo trabajo de clasificación de datos ([RKMG94]). Si bien este trabajo no extiende la clasificación presentada previamente por ellos, es importante destacar la aparición del concepto de conocimiento (*knowledge*) como base fundamental para poder crear una representación gráfica en forma automática.

En 1995 investigadores del IBM Thomas J. Watson Research Center presentaron un artículo ([BRT95]) enfocado en la selección de mapas de colores en forma automática. Este trabajo se dio en el contexto de un trabajo originado en 1993 sobre un sistema de asistencia al usuario en la selección de los parámetros para la configuración de una visualización. Este sistema, denominado PRAVDA (*Perceptual Rule-Based Architecture for Visualizing Data Accurately*), utiliza un sistema de reglas para guiar al usuario durante el proceso de visualización.

Los mapas de colores, de acuerdo a lo presentado en el artículo, dependen principalmente de los datos a visualizar y por esto se caracterizaron los datos de la

siguiente forma: **ratio**, **intervalo**, **ordinal** y **nominal**. Para cada tipo de dato también se analiza la frecuencia espacial del dato, dividiéndola en alta y baja. La combinación de estos dos elementos, tipo de dato y frecuencia espacial determinan las características del mapa de colores (Figura 4.5).

Tipo de dato	Frecuencia Espacial	<i>Isomorphic</i>	Tareas Segmentación	Resaltado
RATIO	Baja	Luminancia: Uniforme. Tono: Pares opuestos o complementarios. Saturación: Monotonicamente creciente desde el gris.	Número de segmentos par. Posiblemente muchos segmentos.	Rango amplio para permitir el resaltado
	Alta	Luminancia: monotonicamente creciente. Tono: Opuestos o complementarios. Saturación: Monotonicamente creciente desde el gris.	Número de segmentos par. Posiblemente pocos segmentos.	Rangos pequeños para permitir el resaltado.
INTERVALO	Baja	Luminancia: Uniforme. Tono: Pares opuestos. Saturación: Monotonicamente creciente desde el gris.	Posiblemente muchos segmentos.	Rango amplio para permitir el resaltado.
	Alta	Luminancia: Monotonicamente creciente desde el gris. Tono: Uniforme o con pequeñas variaciones de tono. Saturación: Monotonicamente decreciente.	Posiblemente pocos segmentos.	Rango pequeño para permitir el resaltado.
ORDINAL	Baja	Luminancia: Uniforme. Tono: Variaciones sobre el disco de tonos. Saturación: Monotonicamente decreciente.	Posiblemente pocos segmentos.	Incremento de luminancia en el área resaltada.
	Alta	Luminancia: Monotonicamente creciente. Tono: Variaciones sobre el disco de tonos. Saturación: Uniforme.	Posiblemente muchos segmentos.	Incremento de saturación en el área resaltada.
NOMINAL	Baja	Luminancia: Uniforme. Tono: Variaciones sobre el disco de tonos.	Menos de 7 segmentos.	Incremento de luminancia o saturación en el área resaltada.
	Alta	Saturación: Uniforme.		

Figura 4.2: Descripción de las características de un mapa de color en base a las características de los tipos involucrados ([BRT95]).

En 1996, M. Zhou y S. Feiner ([ZF96]) presentan una clasificación de datos que puede reconocerse como la más completa hasta el momento. En este trabajo los autores describen los datos a visualizar desde 6 ópticas o dimensiones diferentes. Tales dimensiones son: **tipo**, **dominio**, **atributos**, **relaciones**, **rol** y **sentido**. A cada dato lo denominan **objeto** y cada objeto pertenece a un **dominio** y tiene un **tipo** asociado. Cada objeto también

cuenta con **propiedades** y tiene **relaciones** con otros objetos. El **rol** de un objeto depende de las tareas que el usuario desea realizar y el **sentido** está en función de las preferencias del usuario en cuanto a su interpretación visual. El **tipo** del objeto distingue si el elemento es divisible en partes más pequeñas o no. Se identifican dos tipos de objetos: *atómicos* y *compuestos*.

- *Atómico*: Un objeto atómico es la unidad básica e indivisible de información. Esta característica es dependiente del dominio de aplicación.
- *Compuesto*: Un objeto es compuesto, cuando resulta de la combinación de otros elementos *compuestos* y *atómicos*. Los objetos compuestos, a su vez, se clasifican en conjuntos y estructuras, en función de la relación que tienen los elementos dentro del objeto compuesto. En un conjunto, los objetos que lo componen sólo presentan una relación al conjunto pero el conjunto no impone una relación entre ellos. En el caso de la estructura, la estructura impone una relación entre sus componentes.

El **dominio** de un objeto ubica al mismo en una taxonomía semántica. Cada objeto se puede ubicar en uno de las siguientes dominios.

- *Entidad*. Una entidad es un objeto que existe en forma independiente del resto y posee una identidad única.
- *Concepto*. Un concepto es una idea abstracta que puede existir en forma independiente del resto o debe acompañar a otro objeto. A diferencia de las entidades, los conceptos no tienen una correspondencia física. El concepto abstracto de edad es un elemento independiente de cualquier objeto mientras que la edad de una persona particular es un concepto que debe presentarse junto a la persona en sí, con el fin de indicar su edad.
- *Medición*. Una medición es un valor, numérico o no, que puede estar acompañado de una unidad de medida. Una medida no puede existir por sí sola, sino que debe estar relacionada con un concepto. La medida *120kg*. es una medición cuantitativa del peso de una persona. Una medición también puede ser cualitativa, para el mismo ejemplo del peso de una persona podemos decir que el peso es *alto*.

La tercera dimensión es el **atributo**. Esta dimensión representa características comunes a todas las propiedades de los datos. De acuerdo a sus autores, en esta dimensión se describe la *naturaleza* de los datos. Para cada objeto de dato atómico se consideran 5 atributos dentro de esta dimensión. Para el caso de datos compuestos se incluyen 3 más. Los 5 atributos básicos son: *forma*, *material*, *ubicación*, *dinamismo* e *importancia*. Si el objeto es **compuesto** se incluyen *ordenamiento*, *escalabilidad* y *continuidad*.

- *Forma*. Este ítem indica si un objeto tiene asociada una forma física o no. Una persona o un auto son objetos con una forma física asociada clara. Se consideran tres posibilidades, con forma, amorfo y sin forma. La diferencia entre amorfo y sin forma se basa en que un objeto amorfo puede tener una forma asociada bajo ciertas condiciones pero varía tan frecuentemente que no llega a ser considerado un objeto con forma. Todos los objetos considerados como conceptuales bajo la dimensión de dominio se identifican como objetos sin forma, estos objetos no pueden ser mapeados a un objeto físico pero sí pueden ser representados mediante atributos visuales.
- *Material*. El material de un objeto está determinado por cómo este último interactúa con la luz. Este valor puede ir desde algo tan simple como RGB hasta un modelo complejo de iluminación. Este atributo puede ser utilizado para codificar información.
- *Ubicación*. Mientras que forma y material definen atributos visuales de un objeto, ubicación afecta el dónde y cuándo un objeto será mostrado. La ubicación puede ser espacial o temporal.
- *Dinamismo*. Los objetos pueden ser dinámicos o estáticos. Se considera dinámicos a aquéllos cuya información varía en el tiempo mientras que los estáticos permanecen constantes.
- *Importancia*. Muchas veces, cierta información es más importante que otra. El poder indicar esta característica permite mejorar la selección del mapeo visual y las interacciones sobre la visualización.

Para el caso de los objetos de datos *compuestos* los 3 atributos que se incorporan son *ordenamiento*, *escalabilidad* y *continuidad*.

- *Ordenamiento*. Se identifican dos tipos de ordenamiento dentro de los elementos de un objeto compuesto, cuantitativo y cualitativo. El ordenamiento cuantitativo incluye al ordenamiento numérico y no numérico. El ordenamiento cualitativo es un ordenamiento lógico impuesto por el dominio de aplicación.
- *Escalabilidad*. La escalabilidad de un objeto se clasifica a partir de si los elementos del objeto compuesto representan coordenadas o cantidades.
- *Continuidad*. Determina si los elementos del objeto compuesto son continuos o discretos. Los días de una semana son un conjunto discreto mientras que las temperaturas durante un día son un conjunto continuo.

La cuarta dimensión presentada corresponde a la dimensión de **relación**. Por cada relación se describen las siguientes características: *dependencia funcional*, *constitución*, *atributo* y *enumeración*.

- *Dependencia funcional*. Indica si existe una correspondencia 1 a 1 entre el dominio e imagen de la relación.
- *Constitución*. Indica si una relación representa el concepto de forma-parte-de. Este atributo a su vez se divide en constitución física y conceptual.
- *Atributo*. Si a través de una relación, se indica que un objeto está describiendo características de otro se dice que es una relación de atributo.
- *Enumeración*. Una relación de enumeración describe si un objeto es una extensión de otro o pertenece a otro.

Las últimas dos dimensiones presentadas son **rol** y **sentido**. Como en trabajos anteriores, **rol** se relaciona con los objetivos del usuario y las tareas que desea realizar. En particular se detectan 9 tareas posibles: *categorización*, *agrupamiento*, *identificación*, *diferenciación*, *comparación*, *asociación*, *ranking*, *correlación* y *distribución*. Por el momento no describiremos esta dimensión ya que corresponde a otra sección de esta tesis. La dimensión de **sentido** busca capturar las preferencias del usuario en lo referente a cómo representar cada dato. En el trabajo descrito las posibilidades para la dimensión de sentido fueron limitadas a: *etiqueta*, *lista*, *plot*, *símbolo* y *portrait*.

El trabajo de Zhou y Feiner es sumamente interesante debido al nivel de detalle con el que analizan los datos. Un aporte significativo de esta clasificación es la de considerar los objetos de datos como elementos compuestos, es decir, incorporar la noción de estructura y también de conjunto. Si bien los autores no se refieren al concepto de semántica de los datos, el “significado” de los datos juega un rol fundamental en esta clasificación. Lo único objetable en esta presentación es la composición que se hace en una sola clasificación entre características de los datos y de su representación. En la Figura 4.3 se puede ver un resumen de dicha clasificación.

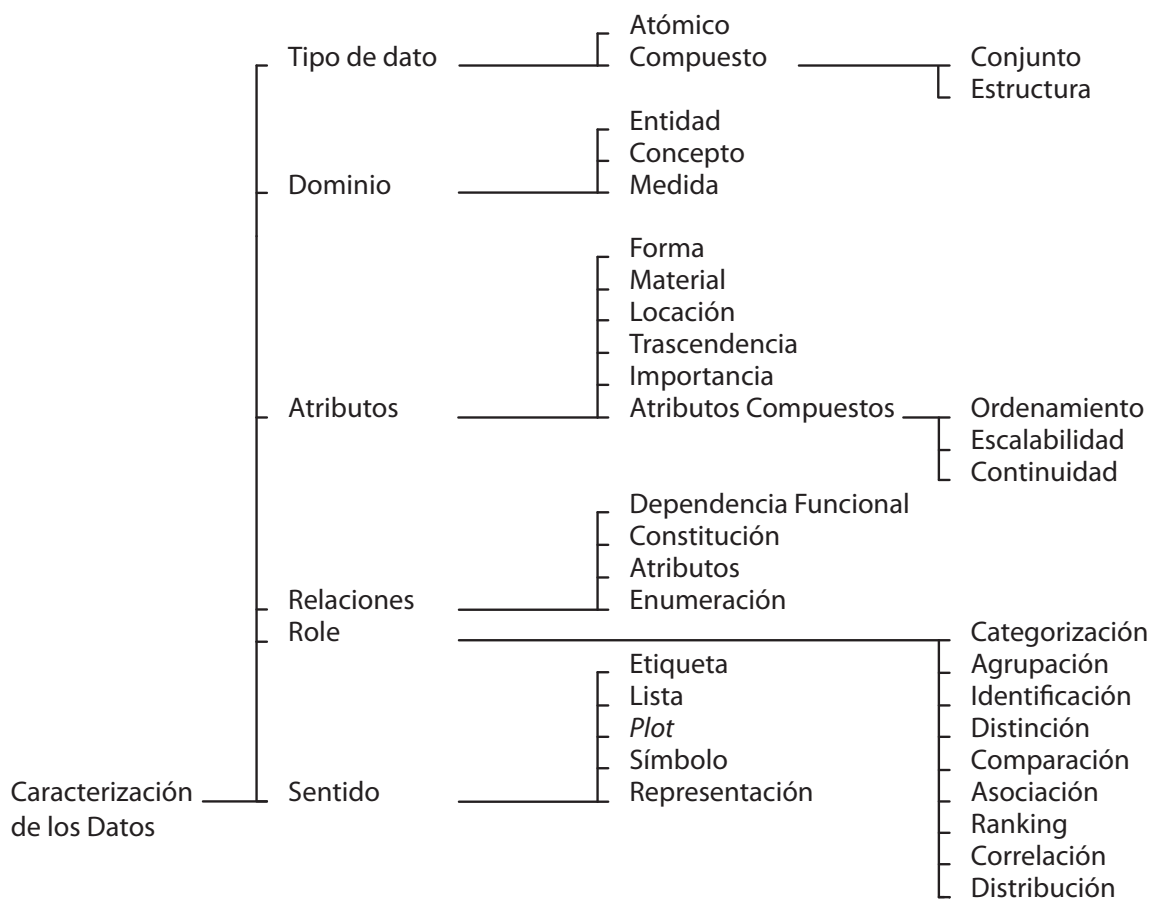


Figura 4.3: Clasificación de datos presentada por Zhou y Feiner ([ZF96]).

Ese mismo año el trabajo de Jiajie Zhang ([Zha96]) presenta un enfoque distinto a los anteriores. En lugar de clasificar datos para generar representaciones de los mismos, comienza clasificando las representaciones para luego derivar la clasificación de los datos. En el contexto de este nuevo enfoque, surgen las dimensiones de una representación.

Una representación visual tiene n dimensiones y cada dimensión es de una escala en particular. Las posibles escalas son **ratio**, **intervalo**, **ordinal** y **nominal**. Ésta es la misma clasificación que ya se vio en los trabajos anteriores, excepto en el de Brodlić ([BCE⁺92]). Cada escala tiene asociado un conjunto de propiedades: *categoría*, *magnitud*, *intervalo igual* y *cero absoluto*.

Categoría se refiere a la propiedad de que las instancias de una escala se pueden distinguir unas de otras. *Magnitud* indica si una instancia en la escala puede ser comparada con otra instancia en la misma escala, la comparación se realiza por mayor, menor o igual. *Intervalo igual* se refiere a la propiedad de que la magnitud de una instancia, representada por una unidad de la escala, es la misma independientemente del lugar en la escala donde se ubique la misma. Finalmente *cero absoluto* indica si la escala asociada permite como valor la ausencia de valor.

También en 1996 Ben Shneiderman ([Shn96]) presenta una taxonomía de tipos de datos para visualización. Éste es el primer trabajo en el que se tienen en cuenta los datos involucrados en una visualización de información. En este trabajo, Shneiderman presenta también una taxonomía de tareas sobre la visualización de información en base a los tipos de datos que se manipulan, siendo éstos: **1D**, **2D**, **3D**, **multi-dimensional**, **temporal**, **árboles** y **redes**.

- **1D**. Los tipos de datos lineales incluyen documentos de texto, código fuente y listados alfabéticos de nombres los cuales son organizados en forma secuencial. Cada ítem de esta colección es una línea de texto que contiene una secuencia de caracteres.
- **2D**. Con este tipo de dato se identifican los datos de planos o mapas, incluyendo mapas geográficos. Cada ítem en esta colección cubre de alguna manera una parte de un área total.
- **3D**. Este tipo de dato incluye objetos de la realidad tales como moléculas, el cuerpo humano o edificios.
- **Multi-dimensionales**. La mayoría de las bases de datos relacionales y estadísticas son manipuladas como datos multi-dimensionales, en donde cada ítem tiene n atributos y es mapeado a un punto en un espacio $n - dimensional$.

- **Temporal.** La distinción entre el tipo de dato temporal y 1D es que los datos temporales tienen un tiempo de inicio y fin. Además, los ítems de esta colección pueden superponerse.
- **Árboles.** Las jerarquías o estructuras de árbol son colecciones de ítems en la que cada ítem se relaciona con otro al que llama padre. Existe uno y sólo un ítem que se denomina raíz y que no posee padre.
- **Redes.** Muchas veces las relaciones entre ítems no pueden ser representadas a través de un árbol y resulta útil permitir enlaces o conexiones arbitrarias entre los ítems. Este es el caso del tipo de dato redes.

El trabajo de Shneiderman es muy significativo por la visión que da de los tipos de datos desde un nivel más alto de abstracción. Sin embargo, este nivel de abstracción no parece suficiente para capturar las características de los datos. Un ejemplo de esto es el tipo **1D** que sólo incluye tipos de datos cualitativos dejando fuera los cuantitativos. Aunque esta clasificación resultó ser muy utilizada, como veremos durante el desarrollo de esta sección, siempre lo fue como complemento a otras clasificaciones. Esta misma clasificación es utilizada nuevamente por Shneiderman junto a Card ([CMS99]).

En 1997 Lisa Tweedie ([Twe97]) realiza una caracterización del espacio de técnicas que existen para las visualizaciones y en el proceso se analizan los tipos de datos que dichas técnicas manipulan. Al igual que en el trabajo de Zhang ([Zha96]) el enfoque que se presenta parte de las técnicas de visualización y, a partir de éstas, deriva las características de los datos.

La autora comienza su clasificación con una división de los datos en **valores** y **estructuras**, similar a lo presentado anteriormente por M. Zhou y S. Feiner ([ZF96]). Tweedie incluye una tercera y cuarta división para los datos a las que denomina **dato construido** y **dato convertido**. Estas nuevas categorías se derivan de la posibilidad de transformar **datos** y **estructuras**. Un tipo de transformación mantiene los **datos** o **estructuras** en su forma original, es decir los valores son transformados en nuevos valores y las estructuras son transformadas en nuevas estructuras. Esta transformación corresponde a **datos construidos**. Otro tipo de transformación cambia la forma de los datos de modo tal que de valores se derivan estructuras o de estructuras se derivan datos. Este caso se denomina **dato convertido**. A partir de esta división Tweedie establece

las formas de representación apropiadas para cada tipo de dato. Este último punto será desarrollado más adelante en esta tesis.

En 1999 Card, Mackinlay y Shneiderman publican el libro “Readings in Information Visualization. Using Vision to Think” ([CMS99]); este trabajo logró convertirse en una lectura fundamental para entender los conceptos y necesidades del área de Visualización de Información. La estructura que presenta este libro en su capítulo 2, *Space*, se corresponde con la división de las técnicas de visualización a partir de las dimensiones que soportan. Esta división coincide, en parte, con la presentada previamente por Shneiderman ([Shn96]). Las divisiones son: **datos físicos**, **1D**, **2D**, **3D**, **nD** (con $n > 3$), **árboles** y **redes**. A diferencia del trabajo anterior de Shneiderman, aquí no se incluye el elemento temporal como una división posible, pero sí se incluyen los **datos físicos**. La principal diferencia entre datos de **1D** y **datos físicos** reside en el dominio de la visualización, los últimos corresponden a una visualización científica mientras que los primeros son para visualización de información. Se incluye, en forma diferenciada de las divisiones anteriores, el concepto de **tipo de dato texto**. En particular, el **tipo de dato texto** es analizado bajo 4 posibilidades: *texto en 1D*, *texto en 2D*, *texto en 3D* y *texto en 3D con tiempo*. Esta última alternativa incluye la visualización de documentos en el espacio tridimensional visualizando también el parámetro tiempo ([Ren94]).

Ed Chi publicó en el año 2000 ([Chi00]) una taxonomía de técnica de visualización utilizando el mismo enfoque que el presentado por Zhang ([Zha96]). En primera instancia se clasifican las técnicas y en función de dicha clasificación se deriva la clasificación de los datos. Tal clasificación incluye: **representaciones de datos geográficos**, **2D**, **multi-dimensionales**, **árboles**, **redes**, **texto** y **web**. Esta descripción presenta muchos puntos en común con el trabajo de Shneiderman ([Shn96]).

En 2002, Melanie Tory y Torsten Möller presentan una nueva taxonomía de visualización ([TM04]) basada en su modelo de datos. Esta taxonomía discrimina los datos en **continuos** y **discretos**. Para los datos **continuos** se consideran las variables independientes, *dimensiones* y las dependientes, *tipos*. Las variables independientes son: *1D*, *2D*, *3D* y *multi-dimensional* (nD). Las variables dependientes son los tipos de datos: *escalares*, *vectores*, *tensores* y *multivariantes*. Los datos discretos son divididos en *conectados*, *grafos* y *árboles*, y *desconectados*, donde nuevamente se separan en *2D*, *3D* y *nD*. Esta taxonomía se detalla en la Figura 4.5.

		Número de variables dependientes			
		1			Muchas
		ESCALAR	VECTOR	TENSOR	MULTIVARIADO
Número de Variables Independientes	1D	Gráfico de líneas			Combinación de los métodos utilizados para escalar, vectorial y tensorial
	2D	Mapas de colores Isolíneas	<i>LIC</i> <i>Trazado de Partículas</i>		
	3D	<i>Rendering</i> de volúmenes Isosuperficies	<i>Glifos</i>	<i>Elipsoides</i>	
	nD	Múltiples vistas de 1D, 2D o 3D			

Figura 4.4: Taxonomía de técnicas de visualización para un modelo continuo ([TM04]).

Conectado		
Visualización de árboles y grafos: Grafos de jerarquía, <i>Space-filling mosaics</i> , diagramas de enlace.		
Desconectado		
Número de variables	2D	<i>Scatter plot</i> , Gráficos de barra
	3D	<i>3D Scatter plot</i> , Gráficos de barra en 3D,
	nD	Múltiples vistas, Agregación, <i>Glifos</i> , <i>Dense pixel displays</i> , Coordenadas Paralelas

Figura 4.5: Taxonomía de técnicas de visualización para un modelo discreto ([TM04]). Esta taxonomía incluye una clasificación de los datos.

La jerarquía presentada en el año 1999 por Card, Mackinlay y Shneiderman es presentada nuevamente, con leves variantes, cuatro años después en el trabajo de Qin Chengzhi, Zhou Chenghu y Pei Tao ([QZP03]). La taxonomía propuesta incluye los siguientes tipos de datos: **1D**, **2D**, **3D**, **multi-dimensional**, **jerarquías**, **grafos** y **texto/hipertexto** (Figura 4.6).

Ese mismo año, Darius Pfitzner, Vaughan Hobbs y David Powers ([PHP03]) presentan una taxonomía en la que incluyen **datos**, **tareas**, **interactividad**, **habilidad** y **contexto**. Por la parte de **datos** se analiza el *tipo de dato* y la *relación de los datos*. Los *tipos de datos* se dividen en *bajo nivel* y *alto nivel*. *Bajo nivel* incluye los objetos y sus atributos y *alto nivel* incluye meta información (Figura 4.7). La relación de los datos contempla la estructura de la relación separándola en *lineal*, *circular*, *árboles ordenados*,

	Tareas Analíticas				
	Overview-query	Comparación	Agrupamiento-Clasificación	Patrones de distribución	Análisis de dependencias y correlación
1D	Animation; LifeLine; line graph; colormap; curve density plo.t	Pie plot; line graph.	Colormap; curve density plot.	Value bars; curve density plot; histogramas	
2D	Mapas geográficos; scatter plot; colormap.	Mapas geográficos; scatter plot.	Colormap	Isogram plot.	AViz
3D	Visible Human; Rendering de volúmenes; scatter plot.	Scatter plot.	Colormap		
nD	Table Lens; n-Vision; Scatterplot Matrix; GrandTour; star glyphs.	Andrews Curve; star glyphs.	Coordenadas paralelas; InfoCrystal; WinViz; HD-Eye	Circle segments; InfoCrystal; GrandTour; Project persuit; FastMap	Scatterplot Matrix Dimension Stacking
Jerarquía	Hyperbolic View; Magic Eye View; Cone Tree; Disk Tree.			Treemap; Information Cube	
Grafo	DA-Tu; Fisheye View; WebBook & WebForage			NetMap	WebView
Texto/Hipertexto	Perspective Wall; Document Lens	TileBars	InfoCrystal	TileBars; InfoCrystal	NetMap

Figura 4.6: Taxonomía de técnicas de visualización ([QZP03]). Esta taxonomía incluye una clasificación de los datos.

árboles no ordenados e hiperespacio. Este último trabajo es el primero en incorporar el concepto de reticulado como tipo de dato.

En 2006 es publicado el trabajo [GHV⁺06], que también enfoca la clasificación a partir de las técnicas y no de los datos ([Zha96], [Twe97]). Este trabajo presenta un dominio de aplicación muy limitado ya que sólo se consideran técnicas 2D, 2 1/2 D y 3D. En el año 2006, Ken Brodlié presenta junto a Nurul Mohd Noor una extensión ([BN07]) de la clasificación presentada anteriormente ([BCE⁺92]). Aquí incorpora cuatro tipos de datos básicos: *reales*, *enteros*, *ordinales* y *nominales*. Se distinguen también los casos donde se tiene un solo valor o un conjunto de valores. La dimensionalidad de los datos también es considerada pero no se hace una categorización a partir de ella como en casos anteriores. Finalmente, en 2008, se presentó comercialmente el sistema *Tableau Software* el cual funciona como un sistema de visualización, análisis y consulta de bases de datos multidimensionales. Este sistema está basado en Polaris ([SH00], [STH02], [STH08]). Originalmente este sistema clasificaba los datos en nominal, ordinal, cuantitativo e intervalo pero luego estos fueron reducidos a ordinal y cuantitativo solamente. La segunda clasificación que se realiza con los datos es la de dividirlos entre dimensiones y medidas, equivalente a los conceptos de variables independientes y dependientes vistos previamente.

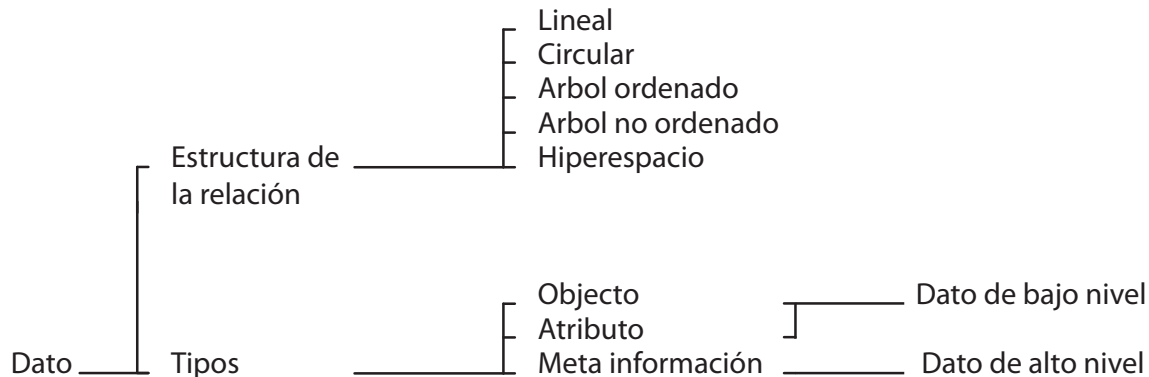


Figura 4.7: Taxonomía de datos presentada por Pfitzner et al. ([PHP03]).

Ninguna de estas clasificaciones puede ser considerada completa debido a que no permiten describir simultáneamente los tipos de datos base y las estructuras, con un nivel de detalle significativo. Hemos presentado clasificaciones que describen los tipos de datos base pero no discriminan los tipos de estructuras, o no las incluyen. En otros casos las estructuras son presentadas detalladamente pero sólo se considera un tipo base atómico, es decir no diferencian, entre *nominal* e *intervalo*.

4.1.2. Nuestra propuesta para la caracterización de datos

Teniendo en cuenta las clasificaciones de datos presentadas anteriormente y considerando que ninguna de ellas considera todos los elementos necesarios a tener en cuenta en el marco de las aplicaciones de visualización en general, proponemos una clasificación de datos desde esta perspectiva. Esta propuesta tiene como objetivo capturar la descripción de los tipos base y de las estructuras, tanto homogéneas y heterogéneas. Adicionalmente esta clasificación facilitará luego la clasificación de los datos de acuerdo a su escalabilidad. Inicialmente se introducen los distintos aspectos a tener en cuenta sobre los datos para clasificarlos, y finalmente se compara esta clasificación con las presentadas anteriormente.

Tomando como base la clasificación presentada por Zhou y Feiner en 1996 ([ZF96]), los tipos de datos permiten distinguir inicialmente si un elemento es o no divisible en otros elementos menores. Por lo tanto, hay dos clases principales de tipos de datos: *atómicos* y *compuestos*. Los datos *atómicos* son aquéllos que representan las piezas básicas de

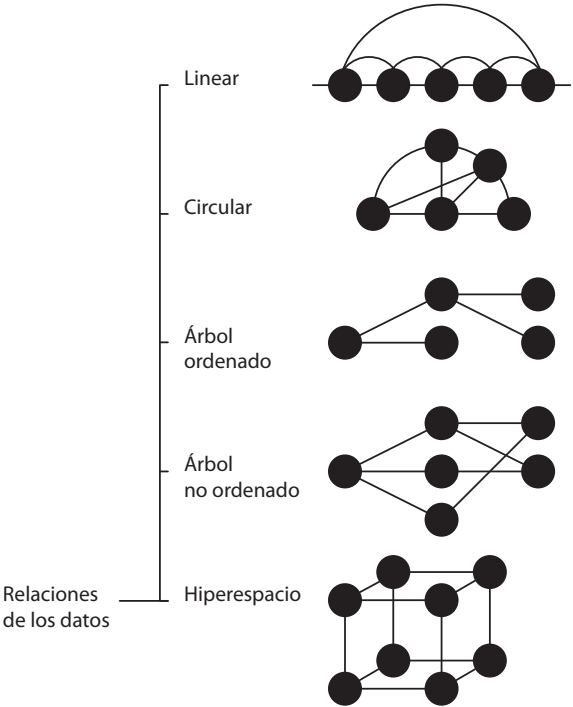


Figura 4.8: Taxonomía de relaciones por Pfitzner et al. ([PHP03].)

DATO				
Atómico		Compuesto		
Discreto	Continuo	Grafo	Heterogéneo	
Ordinal	Ratio	Conjunto	Homogéneo	
Nominal	Intervalo	Secuencia		
	Absoluto	Jerarquía		

Figura 4.9: Nuestra propuesta para la clasificación de los datos.

información. Los datos *compuestos* son aquéllos que representan la combinación de datos atómicos y/o compuestos.

Tanto el conjunto de datos *atómicos* como el conjunto de datos *compuestos* pueden particionarse de diferentes maneras según cuál sea el criterio que se aplica. Dos criterios que utilizamos en este trabajo para particionar los datos *atómicos* son: **la continuidad del dato** y **el tipo de medida que puede definirse sobre el dato**. Para los datos compuestos se utilizan dos criterios diferentes: **la relación existente entre los elementos** y **la homogeneidad de los mismos**.

4.1.3. Datos atómicos

Criterio: Continuidad

Los datos atómicos pueden diferenciarse en datos *continuos* y datos *discretos* ([TM04]). El primer grupo se refiere a datos para los cuales se pueden interpolar los puntos de muestra para predecir valores intermedios. El segundo grupo se refiere a aquellos datos para los cuales la interpolación no tiene sentido.

Criterio: Tipos de escalas de medición

El segundo criterio de división para los datos atómicos es a partir de la medida que puede asociarse a ellos. Los tipos de escalas definidos sobre datos atómicos son los ya presentados ([Ste46], [FP98]): *nominal*, *ordinal*, *intervalo*, *ratio* y *absoluto*.

Dados dos tipos de escala a y b , se dice que a es *más rico* que b si todos los datos que se pueden representar con b , también son representables con a . De esta manera, los tipos de escala fueron listados en orden creciente con respecto a cuán rico es. Sin embargo, cuanto más rico sea el tipo de escala, más restringido es el conjunto de posibles representaciones. Los tipos de datos serán *nominal*, *ordinal*, *intervalo*, *ratio* y *absoluto* dependiendo de cuál es el tipo de escala *menos rico* aplicable sobre los datos. Si M y M' son dos medidas del mismo atributo, entonces se relacionan según la escala del atributo como se muestra en la Figura 4.10.

Tipo de Escala	Relación entre M y M'
Nominal	Mapeo uno a uno entre M y M' .
Ordinal	Si $M(x) > M(y)$ entonces $M'(x) > M'(y)$.
Intervalo	$M' = aM + b, (a > 0)$.
Ratio	$M' = aM, (a > 0)$.
Absoluto	$M' = M$.

Figura 4.10: Se define cómo deben ser dos medidas M y M' para ser medidas válidas sobre cada tipo de escala.

Escala nominal La escala *nominal* es aquella que divide los datos en clases diferentes. Las principales características de la escala *nominal* son:

- La escala sólo cuenta con clases diferentes y no existe ninguna relación de orden entre ellas.
- Cualquier representación simbólica o numérica es una medida aceptable para estas clases, no habiendo ninguna noción de magnitud asociada a esos símbolos o números.

Un ejemplo de medida sobre una escala *nominal* es el siguiente: tenemos un conjunto de datos que pueden tomar los valores *Volkswagen*, *Ford* o *Fiat*. M_1 , M_2 y M_3 son tres posibles medidas aplicables a estos datos:

$$M_1(x) = \begin{cases} \text{VW} & \text{si } x \text{ es Volkswagen} \\ \text{Ford} & \text{si } x \text{ es Ford} \\ \text{Fiat} & \text{si } x \text{ es Fiat} \end{cases} \quad M_2(x) = \begin{cases} 1 & \text{si } x \text{ es Volkswagen} \\ 2 & \text{si } x \text{ es Ford} \\ 8 & \text{si } x \text{ es Fiat} \end{cases}$$

$$M_3(x) = \begin{cases} \circ & \text{si } x \text{ es Volkswagen} \\ \Delta & \text{si } x \text{ es Ford} \\ \diamond & \text{si } x \text{ es Fiat} \end{cases}$$

Notar que aunque M_2 induzca a pensar en la existencia de una relación de orden entre los valores, dado que son datos nominales no existe tal relación.

Escala ordinal La escala ordinal agrega a la escala nominal información de orden entre las clases. Las principales características de la escala ordinal son:

- La escala consiste en clases ordenadas.
- Cualquier mapeo que conserve ese ordenamiento es aceptable.
- Los números sólo representan orden, por lo tanto no tiene sentido la aplicación de operaciones aritméticas sobre los mismos.

Un ejemplo de esta escala es el siguiente: queremos capturar cualitativamente la nota de ejercicios. Inicialmente, se pueden distinguir 5 clases: *mal*, *regular*, *bien*, *muy bien* y *excelente*. Hay una relación de orden implícita *mejor que* entre las notas. Cualquier medida M debe preservar el orden de las notas. Por lo tanto, las siguientes serían medidas válidas:

$$M_1(x) = \begin{cases} 1 & \text{si } x \text{ es mal} \\ 2 & \text{si } x \text{ es regular} \\ 3 & \text{si } x \text{ es bien} \\ 4 & \text{si } x \text{ es muy bien} \\ 5 & \text{si } x \text{ es excelente} \end{cases} \quad M_2(x) = \begin{cases} 1 & \text{si } x \text{ es mal} \\ 2 & \text{si } x \text{ es regular} \\ 3,5 & \text{si } x \text{ es bien} \\ 4 & \text{si } x \text{ es muy bien} \\ 15 & \text{si } x \text{ es excelente} \end{cases}$$

Sin embargo, las siguientes medidas no son válidas:

$$M_3(x) = \begin{cases} 1 & \text{si } x \text{ es mal} \\ 2 & \text{si } x \text{ es regular} \\ 2 & \text{si } x \text{ es bien} \\ 4 & \text{si } x \text{ es muy bien} \\ 5 & \text{si } x \text{ es excelente} \end{cases} \quad M_4(x) = \begin{cases} 1 & \text{si } x \text{ es mal} \\ 5 & \text{si } x \text{ es regular} \\ 3 & \text{si } x \text{ es bien} \\ 4 & \text{si } x \text{ es muy bien} \\ 6 & \text{si } x \text{ es excelente} \end{cases}$$

Escala de intervalo La escala de intervalo es aquella que incorpora información sobre el tamaño del intervalo que separa las clases, y por lo tanto se puede medir el cambio de una clase a otra. Las principales características son las siguientes:

- Hay definido un ordenamiento entre las clases.
- Tiene sentido calcular la distancia entre dos clases cualquiera, pero no la razón entre ellas.

La suma y la diferencia son operaciones aritméticas aceptables sobre una escala definida, pero no así la multiplicación y la división.

Un ejemplo es medir temperaturas en grados Celsius o Fahrenheit. Existe el mismo incremento en calor, pasando de 20 a 25 grados que de 30 a 35 grados. Sin embargo, no se puede decir que si un día hace 10 grados de temperatura y al día siguiente 20 grados, el segundo día hizo el doble de calor que el anterior.

Escala de ratio La escala de ratio agrega la noción de razón entre clases. Tiene las siguientes características:

- Hay definido ordenamiento entre las clases, y tienen sentido tanto el tamaño del intervalo entre clases como la razón entre ellas.

- Hay un elemento cero, que representa la ausencia absoluta del atributo.
- La medida debe empezar en cero y crecer a intervalos iguales (unidades).
- Todas las operaciones aritméticas pueden aplicarse sobre esta clase.

Ejemplos de esta escala son la temperatura medida en Kelvin, longitud e intervalos de tiempo.

Escala absoluta Esta es la escala más restrictiva de todas. Hay una única forma de definir la medida. La escala absoluta tiene las siguientes características:

- La medida de una escala absoluta es simplemente contar el número de elementos.
- Hay una única medida posible, la cuenta real.
- Todas las operaciones aritméticas tienen sentido.

Los ejemplos de la escala absoluta son todas aquellas medidas que implican contar cantidad de entidades.

4.1.4. Datos compuestos

Los datos compuestos son aquéllos que resultan de la combinación de otros elementos *compuestos* y *atómicos*. Los datos compuestos, a su vez, se clasifican de acuerdo al tipo de relación que se establece entre las componentes, por ejemplo si se establece una relación lineal o de jerarquía. Los datos compuestos también se clasifican en función de la igualdad de tipos de sus componentes; si todas sus componentes son del mismo tipo entonces se dice que el dato compuesto es homogéneo; si no, es heterogéneo.

Criterio: relación existente entre los elementos

Según la relación existente entre los elementos, los datos compuestos pueden dividirse en cuatro grandes grupos de datos: conjuntos, secuencias, jerarquías y grafos.

Los *conjuntos* ([ZF96]) son aquellos datos compuestos donde cada componente es considerada sólo en relación al conjunto sin tener en cuenta relaciones con las demás

componentes. Además del posible orden natural existente entre los elementos que componen el conjunto, no hay ninguna otra relación entre ellos más allá que la pertenencia al mismo conjunto.

En las *secuencias*, *jerarquías* y *grafos* existe relación de orden (independiente del posible orden natural que exista) entre los elementos que lo componen.

Un *grafo* consiste en un conjunto de vértices y un conjunto de arcos conectando un subconjunto de vértices. Los vértices del grafo representan objetos y los arcos representan relaciones entre los objetos. Formalmente, un grafo \mathcal{G} [Die05, sec. 1.1] es un par $\langle V, E \rangle$ donde E y V son conjuntos tales que si $x \in E$ entonces $|x| = 2$ y $x \subseteq V$; es decir, los elementos de E son subconjuntos de 2 elementos de V . Los elementos de V son los *vértices* o *nodos* del grafo \mathcal{G} , y los elementos de E son sus *arcos* o *líneas*.

Una *jerarquía* es un conjunto de vértices, de los cuales uno se distingue como raíz, y sobre los que se define una relación en la que cada elemento (excepto la raíz) se relaciona con otro elemento llamado *padre*. Formalmente, una jerarquía \mathcal{T} es un grafo conexo acíclico, es decir que cualquier par de vértices de \mathcal{T} está unido por un único camino.

Una *secuencia* es una jerarquía en la que no hay dos elementos que tengan el mismo padre. De esta manera, establece un orden lineal entre los elementos.

Criterio: diversidad

Según la diversidad de tipo que haya entre los elementos que componen el dato compuesto, estos puede dividirse en homogéneos y heterogéneos. Los datos compuestos *homogéneos* son aquéllos cuyos elementos son todos del mismo tipo de datos, ya sea éste atómico o compuesto. En el caso de datos compuestos por elementos compuestos, estos últimos deben ser homogéneos y tener todos la misma homogeneidad. Formalmente, sea \mathcal{X} un dato compuesto y x_1, x_2, \dots, x_n sus componentes. \mathcal{X} es homogéneo si y sólo si $T(x_i) = T(x_j) \forall x_i, x_j \in \mathcal{X}$ y se cumple una de las dos condiciones siguientes:

1. $T(x_i)$ es algún tipo de dato atómico.
2. $T(x_i)$ es algún tipo de dato compuesto y además $T(x_i)$ es homogéneo $\forall x_i \in \mathcal{X}$ y existen $a \in x_i$ y $b \in x_j$ tal que $T(a) = T(b)$.

donde $T(x)$ representa el tipo de dato de x y puede tomar alguno de los siguientes valores: ordinal, nominal, ratio, intervalo, absoluto, conjunto, secuencia, jerarquía o grafo.

Los datos *heterogéneos* son aquéllos que están compuestos por elementos de diferentes tipos de datos.

4.1.5. Comparación de nuestra clasificación con los antecedentes presentados

El objetivo de esta sección es el de comparar nuestra clasificación con las presentadas en el trabajo previo. Para simplificar la comparación hemos diseñado un ejemplo que cubre los tipos más importantes de nuestra clasificación. Nuestro ejemplo modela un *hospital*; éste cuenta con *médicos* que realizan *operaciones* a *pacientes*. A continuación se muestran los atributos de cada una de estas entidades:

Paciente: Nombre, ID, Edad, Género

Médico: Nombre, ID, Especialidad

Operación: Paciente, Médico, Fecha, Hora, Tipo, Costo

Hospital: Médicos, Pacientes, Operaciones

Clasificación de los datos de acuerdo a nuestra propuesta

Según nuestra clasificación, este conjunto de datos se clasifica de la siguiente manera:

Paciente = Conjunto

- Nombre = Nominal
- ID = ordinal
- Edad = absoluto
- Género = nominal

Médico = conjunto

- Nombre = nominal
- ID = ordinal
- Especialidad = nominal

Operación = Conjunto

- Paciente = conjunto
- Médico = conjunto
- Fecha = secuencia
 1. Día = ordinal
 2. Mes = ordinal
 3. Año = ordinal
- Hora = secuencia
 1. Hora = ordinal
 2. Minutos = ordinal

- Tipo = nominal
- Costo = ratio
- Hospital = grafo
- Médicos = conjunto
- Pacientes = conjunto
- Operaciones = conjunto

Clasificación de los datos de acuerdo a las clasificaciones presentadas

Hemos decidido comparar sólo el ejemplo con las clasificaciones más significativas presentadas. El motivo de esta decisión es el de considerar sólo los aspectos relevantes al desarrollo de esta tesis.

Las clasificaciones propuestas por Stevens ([Ste46]), Mackinlay ([Mac86]), Zhang ([Zha96]) y Cleveland & McGill ([CM84]) no son aplicables a nuestro ejemplo porque no permiten modelar las estructuras de datos. La clasificación presentada por Brodlie ([BCE⁺92]) no da lugar a la representación de aquellos datos cualitativos, por ejemplo *Género*. Corresponde mencionar que aquellos datos como *Género*, *Especialidad* y *Tipo de operación* que toman valores de un conjunto finito de valores, podrían representarse como E_1^S , un valor enumerado de 1 dimensión, por ejemplo género $\in \{\text{femenino, masculino}\}$. Wehrend ([WL90]) et. al nos brindan la primera clasificación que permite clasificar por completo los elementos involucrados en nuestro ejemplo; sin embargo, las estructuras sólo pueden ser descritas como estructuras y no como secuencias o grafos. La siguiente tabla muestra la clasificación de datos para nuestro ejemplo, utilizando la propuesta de Wehrend et. al:

- | | |
|--|---|
| <p>Paciente = Estructura</p> <ul style="list-style-type: none"> • Nombre = Nominal • ID = Escalar • Edad = Escalar • Género = Nominal <p>Médico = Estructura</p> <ul style="list-style-type: none"> • Nombre = Nominal • ID = Escalar • Especialidad = Nominal <p>Operación = Estructura</p> <ul style="list-style-type: none"> • Paciente = Estructura • Médico = Estructura | <ul style="list-style-type: none"> • Fecha = Estructura <ol style="list-style-type: none"> 1. Día = Escalar 2. Mes = Escalar 3. Año = Escalar • Hora = Estructura <ol style="list-style-type: none"> 1. Hora = Escalar 2. Minutos = Escalar • Tipo = Escalar • Costo = Escalar <p>Hospital = Estructura</p> <ul style="list-style-type: none"> • Médicos = Estructura • Pacientes = Estructura • Operaciones = Estructura |
|--|---|

La clasificación presentada por Golemati et. al ([GHV⁺06]) permite clasificar los datos cualitativos, cuantitativos y las estructuras. Sin embargo, dado que se enfoca desde la dimensionalidad de los datos, no define cómo calcular la dimensionalidad de estructuras anidadas; en nuestro ejemplo un caso de estructura anidada es *Hospital*. La clasificación publicada por Pfitzner et. al ([PHP03]) cubre los elementos involucrados en nuestro ejemplo, pero no permite detallar cuáles son los tipos posibles para los datos atómicos. Esta propuesta clasificaría los datos de nuestro ejemplo de la siguiente manera:

Paciente = Data relationship structure.
Ordered Tree.

- Nombre = Data type. Object.
- ID = Data type. Object.
- Edad = Data type. Object.
- Género = Data type. Object.

Médico = Data relationship structure.
Ordered Tree.

- Nombre = Data type. Object.
- ID = Data type. Object.
- Especialidad = Data type. Object.

Operación = Data relationship structure.
Ordered Tree.

- Paciente = Data relationship structure. Ordered Tree.
- Médico = Data relationship structure. Ordered Tree.

• Fecha = Data relationship structure. Ordered Tree.

1. Día = Data type. Object.
2. Mes = Data type. Object.
3. Año = Data type. Object.

• Hora = Data relationship structure. Ordered Tree.

1. Hora = Data type. Object.
2. Minutos = Data type. Object.

• Tipo = Data type. Object.

• Costo = Data type. Object.

Hospital = Data relationship structure.
Un-Ordered Tree.

• Médicos = Data relationship structure. Ordered Tree.

• Pacientes = Data relationship structure. Ordered Tree.

• Operaciones = Data relationship structure. Ordered Tree.

Como se puede ver, las estructuras pueden describirse con un nivel de detalle significativo, pero los tipos atómicos sólo pueden describirse como “*Data type. Object*”. Es posible, dentro de la clasificación de Pfitzner et. al, incluir esta información como parte de la meta información. Algo similar ocurre con la clasificación de Roth y Mattis ([RM90]) pero para las estructuras. En esta clasificación los tipos atómicos se clasifican según sus características pero las estructuras sólo se pueden clasificar como “*Complex Data Types*”. Roth y Mattis contemplan la posibilidad de incluir información extra sobre las estructuras, es decir una descripción de la estructura. Esta información, que cumple el

rol de información semántica, puede ser utilizada para detallar el tipo de estructura que se está representando. Lamentablemente en este trabajo no se detalla el formato de esta información.

4.2. Caracterización de las tareas en Visualización

Como dijimos previamente, entender y conocer la estructura de los datos a visualizar permitirá generar una mejor visualización. Sin embargo, otros elementos también deben ser tenidos en cuenta. Otra característica que se debe considerar es la del objetivo de la visualización. Cuando se inicia el proceso de visualización, se inicia con un objetivo. El primero, y uno trivial, es el de crear una representación visual de los datos. Pero es fundamental preguntarse, ¿por qué se desea crear dicha representación?. Esta razón o razones se denominan, los objetivos del usuario, tareas o *user's goal*. El éxito de una representación visual puede depender del objetivo del usuario. Una técnica de visualización puede facilitar la comparación de datos mientras que otra puede resultar más efectiva para la exploración de los mismos. Por ejemplo, si se desea visualizar una tendencia la mejor técnica es un gráfico de línea mientras que si se desea visualizar proporciones va a ser mejor utilizar un gráfico de torta.

Consideramos que es fundamental definir una caracterización de las posibles tareas a realizar sobre una visualización. Comprender y definir su significado servirá para mejorar el resultado del proceso de visualización. A continuación describiremos diferentes caracterizaciones que fueron presentadas hasta el momento y nuestra propuesta para una caracterización unificada de las tareas a realizar sobre una visualización.

4.2.1. Antecedentes en la caracterización de las tareas

En 1990, Roth y Mattis presentan la primera caracterización de las tareas ([RM90]) en una representación visual, repitiéndose esta caracterización en 1991 ([RM91]). Aunque en estos trabajos no se dan nombres de tareas, sí se identifican los objetivos posibles que tendrán. Tales objetivos, o *énfasis* como se presentan en el trabajo, son:

- *Accurate Value Lookup*. El énfasis se sitúa en la selección de datos.

- *Comparación interna.* El énfasis se encuentra en la comparación de datos dentro de una relación y no entre relaciones.
- *Comparación de pares o n-tuplas.* Aquí se busca comparar relaciones dentro de un mismo conjunto de datos.
- *Distribución de valores en una relación.* El énfasis se sitúa en conocer la frecuencia de ocurrencia de ciertos valores y no la relación entre ellos.
- *Correlaciones entre los atributos.* Tal como se puede ver en un *scatter plot*, el énfasis se encuentra en la comparación de atributos.

Cuatro años después, Keller & Keller ([KK94]) dan una taxonomía de los objetivos de la visualización. En su trabajo, ellos afirman que la identificación de los objetivos de la visualización sugiere cuáles son las técnicas apropiadas a utilizar para alcanzar dicha meta. En ese trabajo, el objetivo de una visualización se define estableciendo primero que tarea se desea realizar y sobre qué tipo de dato. Los autores describen 9 posibles tareas en una visualización, las cuales son:

- *Identificación.* Identificar es establecer las características colectivas por las que un objeto es claramente reconocible. Cuando un objeto individual se está identificando, es cuando se logra el máximo nivel de detalle de la tarea.
- *Localización.* Localizar es determinar una posición específica, puede ser absoluta o relativa. Esta tarea se asocia con frecuencia con una región espacialmente extendida. En estos casos, hay dos propiedades a considerar: ubicación y límites con respecto a la región. Los límites incluyen la magnitud, alcance o distancia sobre la que se extiende.
- *Diferenciación.* Diferenciar es distinguir o reconocer algo como distinto. Esta profundidad de detalle sólo exige que los objetos puedan ser reconocidos como diferentes.
- *Categorización.* Categorizar es colocar objetos en categorías definidas específicamente. La clasificación es a menudo considerada cuando varios objetos son estudiados y algún tipo de organización es requerida.

- *Agrupamiento*. Agrupar es juntar en grupos de iguales, similares, o tipo relacionado. El agrupamiento es utilizado para unir elementos que necesitan ser agrupadas conceptual o físicamente. En la categorización los objetos son agrupados en categorías o grupos ya existentes mientras que en la agrupación los grupos se crean a medida que los objetos se colocan en ellos.
- *Ranking*. Rankear un objeto es darle un orden o posición con respecto a otros objetos de tipo similar. Esta tarea es utilizada cuando se le debe dar una clasificación absoluta o relativa a una serie de objetos.
- *Comparación*. Comparar es examinar con el fin de considerar las semejanzas y diferencias. La comparación es importante cuando dos o más objetos necesitan ser observados, sin implicar una asociación de orden.
- *Asociación*. Asociar es enlazar o unir en una relación dos o más objetos. La asociación se utiliza cuando se establece una relación entre dos o más objetos que de otro modo no se relacionarían. Los objetos involucrados no necesariamente tienen que ser del mismo tipo.
- *Correlación*. Correlacionar es establecer una conexión directa. La correlación se utiliza cuando la relación entre dos o más objetos es sumamente importante.

En 1996, Bergman et al. presentaron un sistema automático ([BRT95]) para la creación de mapas de colores basados en reglas. Dentro del conjunto de reglas que se establecen para describir las características de los colores consideran los tipos de tareas que se van a realizar en la representación visual. Fueron tres los tipos de tareas presentados:

- *Isomorfismo*. Aquí se busca que la visualización represente fielmente la estructura de los datos.
- *Segmentación*. El objetivo de este tipo de tarea es la de dividir los datos en categorías perceptualmente distinguibles.
- *Resaltado*. Aquí se busca destacar una característica de los datos.

En 1996 Jiajie Zhang introduce una nueva taxonomía de tareas ([Zha96]). Con un análisis de alto nivel, Zhang categoriza las tareas en tres grupos, *recuperación de*

información (information retrieval tasks), *comparación* (comparison tasks) e *integración* (integration tasks). Una contribución muy importante por parte de Zhang es la de relacionar las tareas con los tipos de representación. De esta forma se puede ver cómo el significado de una tarea puede definir las características de la visualización.

- *Recuperación de Información.* Este grupo representa aquellas tareas que se centran en la búsqueda de información específica.
- *Comparación.* Este grupo de tareas se enfoca en la comparación de información. Zhang identifica dos tipos de subgrupos: los que corresponden a comparaciones dentro de una misma dimensión y los que corresponden a comparaciones entre dimensiones.
- *Integración.* Este grupo de tareas se centra en aquéllas que requieren que dos o más visualizaciones simultáneas sean integradas mentalmente por el usuario para poder entender lo que se representa. Por el momento no consideraremos esta tarea debido a que esta tesis se enfoca en la creación de una visualización y no de n visualizaciones.

En el mismo año, Zhou y Feiner ([ZF96]) presentaron una caracterización de los datos que incluía no sólo la descripción de la estructura que daba soporte a los datos sino también qué tareas deseaba realizar sobre los datos el usuario. Estas tareas son las mismas presentadas por Keller & Keller ([KK94]). También en este mismo año, Ben Shneiderman introduce su taxonomía de tareas ([Shn96]) que se detalla continuación:

- *Overview.* Tener una vista global de la información.
- *Zoom.* Hacer un acercamiento sobre un ítem de interés.
- *Filtrado.* Remover elementos no deseados.
- *Detalles bajo demanda.* Seleccionar un ítem o grupo de ítems y obtener más información sobre ellos.
- *Relacionar.* Observar relaciones entre ítems.
- *Historial.* Mantener un historial de acciones para luego poder repetir o deshacer.
- *Extracción.* Extraer ítems de colecciones de ellos.

En 2003 Pfitzner et al. ([PHP03]) introducen un framework de trabajo para la visualización de información. Como parte de este framework introdujeron el factor tarea, *task factor*, que permite describir los tipos de tareas posibles o dimensiones del factor tarea. Dichas tareas son las mismas presentadas anteriormente por Zhou y Feiner:

Bongshin Lee et al. introducen, en 2006, una taxonomía de tareas ([LPP⁺06]) para la visualización de grafos. Esta taxonomía está compuesta por 10 tareas y se describen a continuación:

- *Recuperar Valores (Retrieve Value)*. Dado un conjunto de elementos, encontrar ciertos atributos de dichos elementos.
- *Filtrar (Filter)*. A partir de condiciones sobre los valores de los atributos, encontrar elementos cuyos atributos verifiquen esas condiciones. Encontrar casos de datos que cumplan dichas condiciones.
- *Calcular Datos Derivados (Compute Derived Value)*. Dado un conjunto de elementos, computar un valor agregado.
- *Encontrar Extremos (Find Extremum)*. Encontrar elementos que posean valores extremos en sus atributos.
- *Ordenar (Sort)*. Dado un conjunto de elementos, rankearlos de acuerdo a una métrica numérica.
- *Determinar Rango (Determine Range)*. Dado un conjunto de elementos y un atributo de interés, buscar para éste el espacio de los valores dentro del conjunto.
- *Caracterizar Distribución (Characterize Distribution)*. Dado un conjunto de casos y un atributo cuantitativo de interés, caracterizar la distribución de los valores del atributo en el conjunto.
- *Encontrar Anomalías (Find Anomalies)*. A partir de una relación dada o expectativa, identificar cualquier anomalía dentro de un conjunto de elementos.
- *Agrupar (Cluster)*. Dado un conjunto de elementos, agruparlos a partir de valores similares en sus atributos.
- *Correlacionar (Correlate)*. Dado un conjunto de elementos y dos atributos, determinar las relaciones útiles entre los valores de esos atributos.

Finalmente, en 2009, Breslow et. al ([BRT09]) indican que un diseñador de una visualización debe ser guiado por las tareas que se aplicarán sobre la visualización. Enfocándose sólo en *color-coded visualizations*, Breslow et. al identifican dos tipos de tareas: *Identificación y Comparación*. Ambas tareas son descritas de igual forma que en el trabajo de Keller & Keller ([KK94]).

4.2.2. Nuestra propuesta para la caracterización de tareas

Antes de comenzar con nuestra propuesta, es importante destacar una diferencia significativa en los antecedentes descritos anteriormente. En la subsección anterior podemos identificar dos grupos de tareas, aquellas de un nivel alto de abstracción como *Correlacionar, Comparar y Asociar*, y otras de más bajo nivel como *Zoom y Overview*. El primer grupo de tareas representa las acciones que el usuario *quiere hacer*, el motivo por el cual inicia el proceso de visualización, e.g. se quieren visualizar dos columnas de una tabla en una base de datos para observar la correlación que existe entre ellas. El segundo grupo de tareas identificado representa las acciones que el usuario *tiene que hacer* para poder lograr lo que *quiere hacer*, i.e. las tareas de bajo nivel dan soporte a las de más alto nivel de abstracción. Por ejemplo, para poder correlacionar las dos tablas que se están visualizando el usuario podría hacer un *overview*, después un *zoom* y finalmente un *filtrado*.

Desde nuestra interpretación del concepto de **tareas del usuario**, las tareas que nos interesa describir en este momento, son aquéllas de un nivel de abstracción alto. Actualmente el Lic. Sergio Martig, miembro del VyGLab, está trabajando en la definición de las tareas de bajo nivel, que se consideran las primitivas de ejecución del MUV. El primer grupo de tareas identificado será denominado *tareas de nivel intermedio* y el segundo *tareas de nivel bajo*. El mayor nivel de abstracción se encuentra constituido por tres tareas ([Kov04]):

- **Ilustrar.** Ilustrar significa mostrar; visualizar para ilustrar implica solamente mostrar los elementos que se desean visualizar.
- **Razonar.** Razonar significa entender; visualizar para razonar implica generar sobre el usuario un proceso de entendimiento, a partir de la visualización, de por qué esos elementos se están visualizando.

- **Descubrir.** Descubrir significa hallar; visualizar para descubrir implica visualizar un conjunto de datos, y a partir de dicha visualización hallar aquellos elementos que son relevantes.

Nuestra caracterización no utilizará estas tareas debido a que su alto nivel de abstracción las hace difíciles de caracterizar. En su lugar utilizaremos aquellas de nivel intermedio. Las interacciones de nivel intermedio sirven para implementar aquellas de más alto nivel de abstracción. Para nuestro trabajo, partimos de la propuesta presentada por Keller y Keller ([KK94]). A esta propuesta le incorporaremos una nueva tarea, la de *explorar*. La necesidad de incorporar esta nueva tarea radica en que en las presentadas por Keller & Keller todas las tareas tienen objetivos específicos y al momento de ejecutarlas se ejecutarán sobre datos específicos. Sin embargo, debemos poder contar con la posibilidad de ver los datos sin aún establecer sobre un subconjunto de los mismos una tarea particular. La tarea *explorar* busca completar ese espacio. Una segunda diferencia entre nuestra propuesta y la de Keller & Keller es la definición de *Correlacionar*. Consideramos que *Correlacionar* es una tarea necesaria pero no bajo la definición provista en el trabajo previo. Bajo nuestra interpretación *Correlacionar* es establecer un grado de semejanza entre dos o más objetos; en el trabajo previo no se considera la posibilidad de tener grados de semejanza, algo que creemos es necesario especificar. Si bien este conjunto de tareas es adecuado para la visualización, queda aún por demostrar si esta clasificación es apropiada para cualquier dominio de aplicación.

4.3. Caracterización del contexto en Visualización

El contexto describe el ambiente en el cual ocurre la visualización, incluye el hardware de base, el dominio de aplicación y el ambiente físico. Poder caracterizar el contexto servirá para construir una representación semántica del contexto en el cual se construye y se utiliza una visualización.

4.3.1. Antecedentes en la caracterización del contexto

En el ámbito de la Visualización de Información el único antecedente encontrado fue en el trabajo de Golemati et al. ([GHV⁺06]) en donde se establece que la efectividad de

una visualización puede depender del *hardware* disponible y de los dispositivos periféricos conectados. Como ejemplo se indica que las características de los dispositivos de salida pueden afectar la elección de la técnica de representación. Aunque una técnica de representación 3D puede ser utilizada con un monitor 2D ciertamente se logrará una mejor percepción de la profundidad si se cuenta con un dispositivo de salida 3D. En este trabajo se denomina *System Context Properties* a las propiedades del sistema actual. Son 3 las categorías que forman este contexto y cada una de ellas indica si cierto dispositivo se encuentra disponible:

- **Dispositivos de Entrada.**

- *Ratón.*
- *Teclado.*
- *Joystick.*
- *Dispositivos especiales (3D mouse, guante virtual, etc.).*

- **Dispositivos de Salida.**

- *Monitor 2D.*
- *Monitor 3D.*
- *Dispositivo de realidad virtual.*

- **Otro Hardware.**

- *Procesador.*
- *Memoria.*
- *Gráficos.*

Esta caracterización es muy significativa debido a ser el único antecedente disponible. Sin embargo es claramente acotado el alcance de representación que tiene. Hoy en día muchos dispositivos pueden ser interpretados como un teclado; por otro lado, el teclado de una computadora y el teclado de un celular son sin duda teclados pero con características totalmente diferentes. Al momento de mapear interacciones como acciones con un teclado, el diagramado y el tamaño del mismo son características que se deben conocer.

Aunque el trabajo de Golemati et al. sea el único antecedente encontrado en el área de Visualización de Información, el concepto de contexto es frecuente en el área de Interacción Humano Computadora (IHC) ([GC00]). En IHC se considera que el contexto no sólo está constituido por el hardware de base sino también por el ambiente físico y social en el que ocurre la interacción entre el humano y la computadora. Robert Bailey le dedica dos capítulos completos, titulados *Physical and Social Environment* y *Input and Output Devices* a este tema en su libro ([Bai96]) sobre diseño de interfaces profesionales. En *Input and Output Devices*, Bailey describe la importancia de conocer tanto las características del *display* que se utilizará como las características del ambiente en donde se ubica dicho *display*. En el capítulo *Physical and Social Environment* se describen las condiciones ideales del ambiente para diferentes tareas. Se incluyen aquí características referidas al calor, frío, brillo y ruido, diferenciando este último entre continuo, intermitente y aquel que interfiere con la comunicación vocal. Ben Shneiderman también incluye las características del *hardware* en su libro ([Shn97]) sobre diseño de interfaces de usuario; en éste se refiere a los dispositivos de interacción, teclados, dispositivos de puntero, audio, *displays* e impresoras.

No existen, al momento del desarrollo de esta tesis, publicación alguna que considere el dominio de aplicación y el dominio organizacional en el proceso de visualización. Por un lado esto resulta negativo ya que no existe un trabajo previo validado que sirva como base pero por otro esta situación manifiesta la necesidad de dicha caracterización y afirma el aporte de esta tesis.

4.3.2. Nuestra propuesta para la caracterización del contexto

Nuestra propuesta es incorporar las características del ambiente en donde ocurre la visualización. Por ambiente denotamos el ambiente físico, el ambiente de aplicación o dominio de aplicación, el dominio organizacional y el hardware de base. Las características del ambiente físico incluyen la luz ambiental, sonido ambiental y distancia del usuario a la visualización. Las características del dominio de aplicación serán conceptos propios del dominio y cómo se mapean a características de la visualización. El dominio organizacional se representará de igual forma que el dominio de aplicación. El hardware de base representa el hardware presente en el dispositivo que está utilizando el humano para ejecutar el proceso de visualización o interactuar con la visualización.

Nuestra propuesta para la caracterización del ambiente físico

Si bien nuestra propuesta para esta caracterización incluye los elementos descritos por Bailey ([Bai96]), se considera una clasificación diferente de los mismos y se completa con nuevos elementos que se consideraron necesarios para la visualización. El ambiente físico se describe como una composición de 5 elementos: luz, sonido, temperatura, usuario-dispositivo.

1. **Luz:** La descripción del parámetro luz en el ambiente se desdobra en dos partes; por un lado se mide **intensidad** de la luz en el ambiente y por otro **reflejo**. Este último parámetro mide la intensidad de la luz directamente sobre el *display*.
2. **Sonido.** El análisis del sonido en el ambiente se realiza en tres partes. Se considera el grado de **sonido continuo**, **sonido intermitente** y **sonido de interferencia**. El **sonido de interferencia** es aquél que interfiere con la comunicación verbal, tal como lo describió Bailey.
3. **Temperatura.** Este parámetro representa la temperatura ambiente.
4. **Usuario-dispositivo.** Con este parámetro buscamos representar aquellas características del ambiente específicas de la relación que se genera entre el usuario y el dispositivo sobre el cual se está desarrollando o utilizando la visualización. Por el momento sólo incluiremos la **distancia al display**, pero actualmente estamos analizando qué otros elementos se pueden incorporar a este parámetro para mejorar la descripción del contexto.

Nuestra propuesta para la caracterización del hardware

Nuestra propuesta para la caracterización del hardware se basa en la presentada por Golemati et al. ([GHV⁺06]) y es extendida con nuevos elementos. Nuestra caracterización del *hardware* se divide en 5 categorías: *Entrada*, *Salida*, *GPU*, *CPU* y *Otros*. A continuación detallamos el rol de cada categoría y su contenido.

- *Entrada.* Representa los elementos utilizados como entrada de información al sistema. Aquí se considera si se cuenta con:
 - *Mouse 2D.*

- *Mouse 3D*.
- *Teclado*.
- *Joystick*.
- *Guante de Realidad Virtual*.
- *Superficie Sensible al Tacto*.

Cada una de estos ítems es un elemento booleano que indica sí existe o no el dispositivo. A su vez, para cada uno de ellos se considera la posibilidad de incluir una descripción más específica del dispositivo como marca, modelo y velocidad de transmisión.

- *Salida*. Representa los elementos utilizados como salida de información del sistema. Aquí se considera si se cuenta con:

- *Monitor 2D*.
- *Monitor 3D*.
- *Head Mounted Display*.
- *Proyector*.
- *Sistema de Sonido*.

Cada uno de estos ítems indica si se incorpora o no el dispositivo. También se considera el tamaño del dispositivo de salida y la cantidad de ellos. Al igual que en el caso anterior se incluirá una descripción específica del dispositivo.

- *GPU*. La unidad de procesamiento gráfico es fundamental para poder utilizar muchas de las técnicas de representación actuales. No solo es necesario crear una representación gráfica sino también contar con tiempos de respuesta razonables ante las interacciones del usuario. Aquí se consideran los siguientes elementos:

- *Número de GPUs*.
- *Versión de DirectX Soportada*.
- *Version de OpenGL Soportada*.
- *Procesador*.
- *Memoria*.

- *CPU*. La capacidad de procesamiento general del sistema es también un dato fundamental al momento de determinar el volumen de información que el sistema puede manejar. En esta categoría se consideran los siguientes elementos:
 - *Número de CPUs*.
 - *Tipo de CPU*.
 - *Velocidad del Reloj*.
- *Memoria*. La cantidad de memoria presente en el sistema también determinará el volumen de datos que el sistema pueda manejar en un determinado momento. En esta categoría se incluye la cantidad de memoria y el tipo de memoria que se utiliza:
 - *Cantidad de Memoria*.
 - *Velocidad de la Memoria*.

Nuestra propuesta para la caracterización del dominio de aplicación

El objetivo de caracterizar el dominio de aplicación es el de describir cómo el ambiente de aplicación afecta las decisiones involucradas en el proceso de visualización. Esta caracterización se basará, al igual que la del modelo cultural, en un mapeo de conceptos a características de la visualización. La relación entre conceptos y colores, así también como conceptos y otras características no solo depende del modelo cultural sino también del área de aplicación de la visualización. Existen mapeos de conceptos a colores que son particulares de ciertos dominios, como la química o la ingeniería civil. La caracterización del dominio de aplicación no forma parte de la caracterización del proceso de visualización debido a que este último se basa en el MUV, un modelo de visualización que tiene como uno de sus objetivos ser independiente del dominio de aplicación.

Nuestra propuesta para la caracterización del dominio organizacional

Al analizar la organización (empresa, negocio, etc.) en la cual se ubica un humano ([JS03]) podemos comprender mejor cómo el mismo interactuará con una aplicación y qué tipo de diseño será más benéfico. Las características de la organización también afectarán las características de la visualización; una empresa puede querer, por ejemplo, que se utilicen los colores institucionales en la representación visual de ciertos datos. Al

igual que con el modelo cultural y el dominio de aplicación, el dominio organizacional funcionará como un mapeo de conceptos a características visuales.

4.4. Caracterización del humano en Visualización

El humano es un elemento esencial en el proceso de visualización. Es quien desarrolla y crea la representación, denominado “el desarrollador” y también quien la usa, denominado “el usuario”; el desarrollador y el usuario se corresponden en la mayoría de los casos a la misma persona. El usuario es quien interactuará con la visualización. Sin embargo, son pocos los trabajos en los que se hayan analizado las características de dicho usuario y cómo influyen el proceso de visualización. A continuación presentaremos un relevamiento de los trabajos que caracterizan al usuario y al desarrollador en el contexto de la visualización. Este tipo de caracterización en Visualización tienen muy pocos antecedentes. Sin embargo, en las áreas de Web Semántica e Interacción Humano Computadora existen trabajos significativos de caracterización del humano. Debido a que estas dos áreas están estrechamente ligadas con el trabajo de esta tesis, se han incluido dichos trabajos en los antecedentes. Concluiremos esta sección con la presentación de nuestra caracterización del usuario.

4.4.1. Antecedentes en la caracterización del humano en Visualización

En 2002 y 2004, Tory y Möller presentaron trabajos en donde se incluía el modelo conceptual del usuario ([TM02]), y que posteriormente se denominó *modelo de usuario* ([TM04]). Este modelo representa los preconceptos que el usuario tiene y cómo interpreta la información. Aunque en dichos trabajos el modelo de usuario se relaciona más directamente con la semántica de los datos, es importante destacar la inclusión del usuario como parte del proceso de visualización.

En 2006, Golemati et al. ([GHV⁺06]) presentan una descripción del usuario denominada *user context properties*. Esta descripción incluye no sólo el nivel de educación del usuario sino también sus características generales y habilidades. A *nivel de educación* se diferencia primaria, secundaria o de grado. Como *información general* se incluye la edad, el género y la profesión. En el área de *habilidades* se describen la capacidad de memoria

visual, la memoria aritmética y el reconocimiento de color. Los autores también optaron por incluir la capacidad del usuario para la recuperación de información (*information retrieval knowledge*), calificándola como novato, medio y perfecto.

4.4.2. Antecedentes en la caracterización del humano en la Web Semántica

En el contexto de la Web Semántica se han realizado múltiples trabajos ([RK07], [GKV⁺07], [CGC07] y [SEZJM08]) con el objetivo de caracterizar al usuario; muchos sólo llegan a un nivel de taxonomía mientras que otros alcanzan el nivel de ontología.

*FOAF*¹ es una ontología para redes sociales creada en el año 2000. El objetivo de *FOAF* es representar, en un lenguaje entendible para las computadoras y los humanos, una descripción de las personas, sus propiedades y sus vínculos sociales (Figura 4.11).

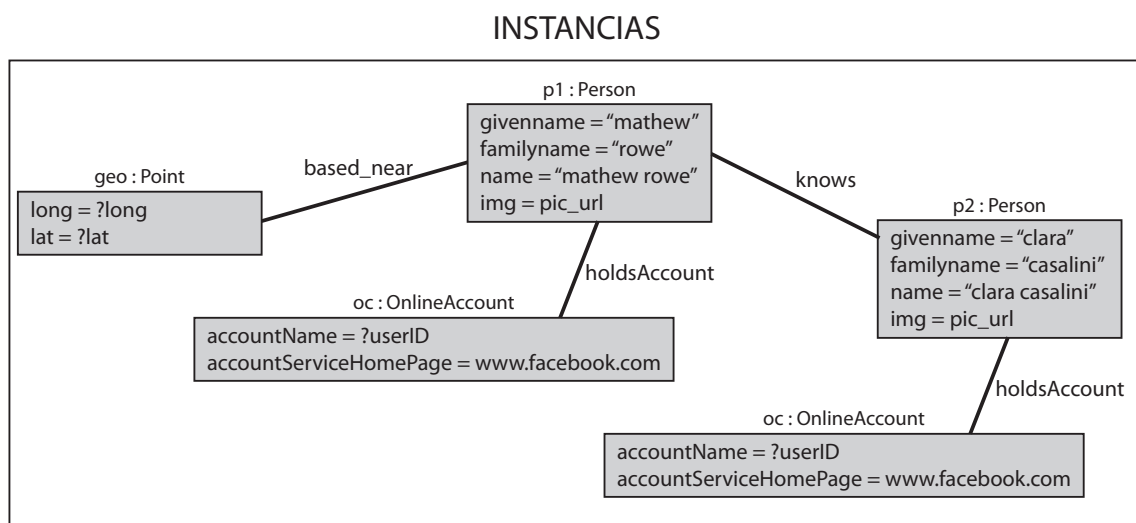


Figura 4.11: Gráfico de relaciones entre instancias en la ontología de *FOAF*.

El mayor aporte de *FOAF* es su vocabulario, un espacio RDF/XML con elementos para describir las características antes mencionadas. *FOAF* fue desarrollado por el *World Wide Web Consortium (W3C)* y su enfoque se centró en la descripción de las propiedades de los seres humanos, es decir, edad, fecha de nacimiento, locación y de sus relaciones sociales (Figura 4.12).

¹<http://www.foaf-project.org/>

FOAF Basics	Personal Info	Online Accounts	Projects/Groups	Documents
Agent	weblog	OnlineAccount	Project	Document
Person	knows	OnlineChatAccount	Organization	Image
name	interest	OnlineEcommerceAccount	Group	PersonalProfileDocument
nick	currentProject	OnlineGamingAccount	member	topic (page)
title	pastProject	holdsAccount	membershipClass	primaryTopic
homepage	plan	accountServiceHomepage	fundedBy	tipjar
mbox	based_near	accountName	theme	sha1
mbox_sha1sum	workplaceHomepage	icqChatID		made (maker)
img	workInfoHomepage	msnChatID		thumbnail
depiction	schoolHomepage	aimChatID		logo
(depicts)	topic_interest	jabberID		
surname	publications	yahooChatID		
family_name	geekcode			
givenname	myersBriggs			
firstname	dnaChecksum			

Figura 4.12: Clases y propiedades utilizadas en *FOAF*.

En [GKV⁺07] se extiende el contenido de *FOAF* y se introducen elementos como *ability* para describir las capacidades del humano (Figura 4.13). En este último caso, por ejemplo, se consideran las personas con problemas visuales, en particular daltonismo. En la Figura 4.14 vemos la incorporación de información contextual del usuario, como nivel de educación e *Information Retrieval Knowledge* así también como la extensión del item *Abilities*.

En el trabajo realizado por Robal and Kalja ([RK07]) se presenta un sistema para la generación de recomendaciones para la adaptación de página web, basándose en la información de localidad del usuario y la ontología del sitio web. Challam et. al desarrollaron un acercamiento a la personalización de los resultados de los motores de búsqueda utilizando perfiles de contexto basados en ontologías ([CGC07]). Este trabajo se centró en la construcción de perfiles de usuarios que capturan lo que el usuario está trabajando en el momento de realizar una búsqueda. Stan et. al ([SEZJM08]) presentaron una ontología basada en el modelo de perfil de usuario, que permite a los usuarios en una red social modificar su estado de disponibilidad en forma automática. La Figura 4.15 muestra un fragmento de la ontología *Situation-Aware User Profile Ontology* (SAUPO).

Clases de la ontología <i>User Profile</i>	
Nombre de la clase	Descripción
Persona	Información básica del usuario: nombre, fecha de nacimiento, e-mail.
Características	Características generales del usuario: color de ojos, altura, peso, etc.
Habilidades	Habilidades y discapacidades, tanto mentales como físicas.
Condiciones de vida	Información relevante relacionada con el lugar y tipo de residencia del usuario.
Contactos	Otras personas con quien el usuario tiene contacto, se incluyen parientes, amigos y compañeros.
Preferencias	Preferencias del usuario, tales como “le gustan los perros”, “le gusta el color azul” o “no le gusta la música clásica”.
Intereses	Hobbies e intereses relacionados con el trabajo.
Actividades	Actividades que realiza el usuario, tanto como hobby o por trabajo.
Educación	Grado de estudios. Diplomas e idiomas.
Profesión	La profesión del usuario

Figura 4.13: Descripción de una persona a través de la ontología *User Profile*, tal como es presentada en el trabajo de Golemati et. al ([GKV⁺07]).

4.4.3. Antecedentes en la caracterización del Humano en HCI

En el área de HCI se identifican tres elementos involucrados en la interacción entre un humano y una computadora: el contexto, la actividad y el humano, siendo este último el más complejo de los tres. La performance del humano puede verse afectada, positiva o negativamente, por una serie de condiciones o influencias que existen en forma independiente de la actividad que se desarrolla o el contexto donde se desarrolla la misma ([Bai96]). Un diseñador debe entender las posibles causas de deficiencias en las personas y tenerlas en cuenta a la hora de tomar decisiones. Por ejemplo, utilizar códigos de colores en las pantallas es común en muchos sistemas. Sin embargo, la codificación por color no debe ser la única forma de codificación, porque muchas personas presentan algún tipo de daltonismo.

Las consideraciones más importantes del elemento humano como un sistema son: los sensores, el procesamiento (cognitivo) del cerebro, y los actuadores (Figura 4.16). La

Propiedades del contexto del usuario	
Propiedades del Contexto	Valores
Educación	Primaria Secundaria De grado
Título, role o relación con la universidad	Miembro de la universidad Personal administrativo Estudiante Ninguna de las anteriores
Capacidad para la recuperación de información	Perfecta Media Baja
Objetivo	Investigación Publicación Información personal
Edad	Entero
Género	Hombre/Mujer
Profesión	Instancia de la clase <i>Profession</i>
Ambiente en donde vive	Instancia de la clase <i>Living environment</i>
Habilidades	Memoria visual Memoria aritmética Reconocimiento de color Rotación mental Habilidades motoras

Figura 4.14: Descripción de la información contextual asociada a una persona según Golemati et. al ([GKV⁺07]).

gente integra una amplia gama de habilidades básicas a una actividad. Estas incluyen una buena visión y una audición adecuada (sensores), brazos, dedos y una boca (actuadores) y la capacidad de pensar, razonar y tomar decisiones (procesamiento cerebral). Tratar de diseñar un sistema sin tener una buena comprensión de cómo las personas sensan, responden y procesar la información es como tratar de cablear una casa sin entender los principios básicos de la electricidad. En ambos casos se puede hacer, pero los resultados pueden ser muy pobres.

En lo que respecta a la disminución del rendimiento de un humano. ésta podría suceder si alguna de las capacidades básicas necesarias para realizar una actividad no existe o se encuentra reducida. Se espera una performance degradada si, por ejemplo, una persona no ha tenido un tiempo de sueño adecuado, tiene una visión menor a una visión perfecta, no ha aprendido ciertas habilidades básicas, o no desea llevar a cabo la actividad requerida.

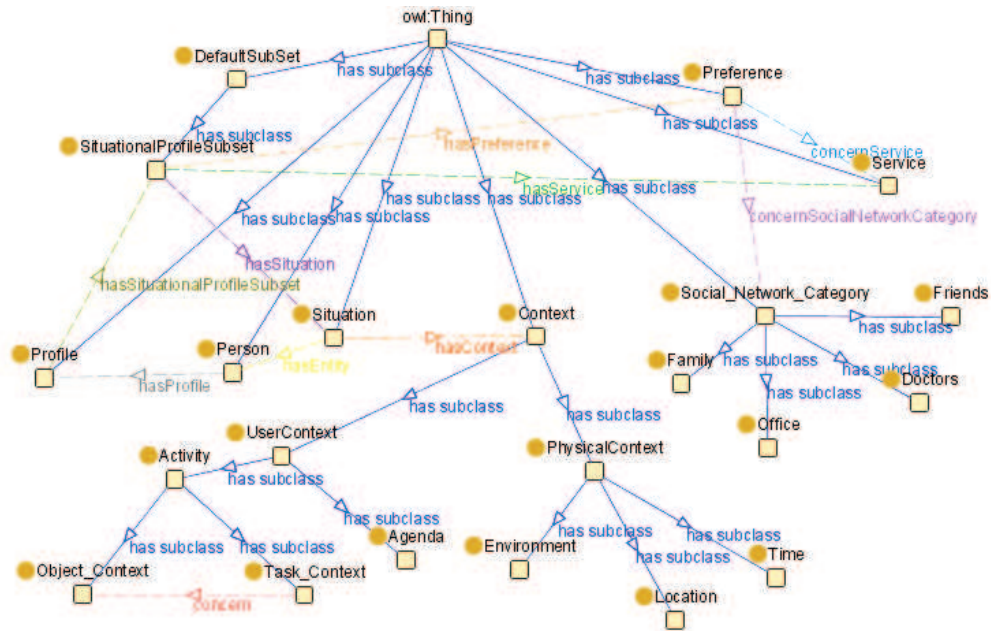


Figura 4.15: Ontología basada en el modelo de perfil de usuario ([SEZJM08]). Permite a los usuarios en una red social modificar su estado de disponibilidad en forma automática.



Figura 4.16: Los elementos considerados en el humano, visto como un sistema ([Bai96]).

Los diseñadores de sistemas suelen hacer varias suposiciones sobre los usuarios de sus sistemas:

- Los usuarios tienen ciertas habilidades, como hablar, escuchar, escribir y utilizar un teclado/mouse.
- Los usuarios tienen las habilidades cognitivas adecuadas, tales como la capacidad de percibir, tomar decisiones, resolver problemas, y controlar el movimiento.

- Los usuarios tienen el deseo o motivación para realizar la actividad requerida. Incluso si la persona tiene las habilidades básicas y ha adquirido las habilidades necesarias, la cuestión sigue siendo si la persona llevará a cabo la actividad o no.
- Los usuarios están mentalmente sanos. Por ejemplo, no se ven presionados o no tienen un alto nivel de ansiedad crónica, no tienen una fobia que afecta su rendimiento, no son compulsivos en sus reacciones, y no tienen obsesiones en relación al desempeño de una actividad.

Los diseñadores de sistemas deberían saber que estas suposiciones no siempre son verdad. Algunos usuarios pueden no tener las habilidades necesarias para realizar cierta actividad, o no tener las capacidades requeridas para aprender las habilidades que la actividad requiere. Debido a que la mayoría de los sistemas son diseñados para grandes grupos de personas, es necesario considerar posibles fallas o debilidades en el humano, particularmente en los sensores, el procesamiento del (cognitivo) cerebro, y los actuadores (Figura 4.16). Estos elementos se integran bajo el “Modelo de Procesamiento de Información” ([Bai96]).

En el contexto de HCI, se considera que el tratamiento de la información por el cerebro se realiza en términos de etapas. La información ingresa a partir de los sensores y llega a una **etapa perceptual**. Una vez tratada, la información pasa a una **etapa de traducción** donde se traduce de la percepción a la acción. Finalmente la última etapa es la **etapa de control de movimiento**. Se considera que cada etapa tiene acceso a la memoria de la persona.

Modelo de Procesamiento de Información

El modelo presentado por Robert Bailey ([Bai96]) considera tres etapas de procesamiento dentro del cerebro: **percepción**, **intelección** y **control de movimiento**. Este modelo de etapas (Figura 4.17) representa el proceso cognitivo que todo diseñador de sistemas debe considerar. Tal como se presentó anteriormente, las tres etapas hacen uso de la memoria. Los sensores (Figura 4.17) se dividen en dos grupos, externos e internos. Los sensores externos son aquéllos que perciben información externa, como por ejemplo los ojos, la nariz y la boca. Los internos perciben la información que proviene del interior del cuerpo humano y pueden subdividirse en dos; aquéllos presentes en los músculos,

los tendones y las articulaciones y los que sensan información relacionada con la sangre, la deshidratación de cuerpo y otros elementos relacionados al estado del cuerpo. Los actuadores (Figura 4.17) también se dividen en dos grupos. El primer grupo corresponde a aquellas partes del cuerpo que se controlan con movimientos voluntarios, tales como las mano, los pies y las cuerdas vocales. El otro grupo consiste de elementos sobre los cuales no hay un control directo voluntario, como por ejemplo las glándulas, el corazón y los órganos internos en general.

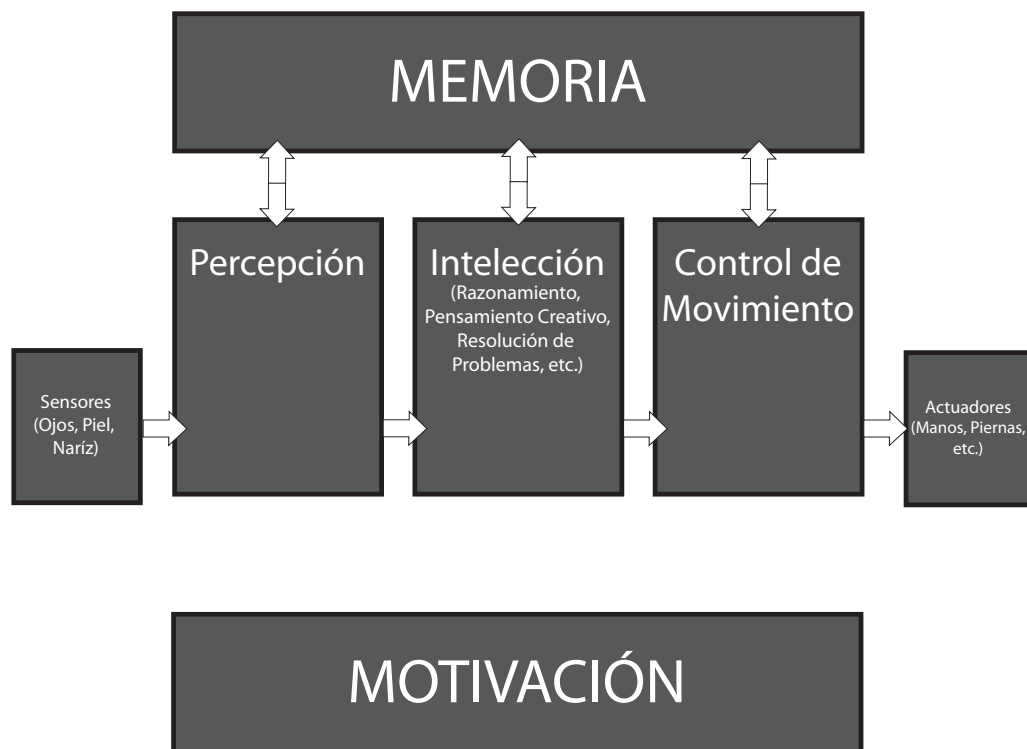


Figura 4.17: Modelo de Procesamiento de Información ([Bai96]).

Supongamos que una persona abre un cajón de su escritorio y encuentra una serpiente lista para atacar; la reacción ante este evento presenta claramente los tres estados antes descriptos. Primero se debe *reconocer* al animal, luego se debe *decidir* que es un animal peligroso y que se debe cerrar el cajón inmediatamente y finalmente se debe ejecutar dicha decisión. Un error en cualquiera de las tres etapas podría provocar un resultado negativo. Si la persona confunde el animal con un juguete e intenta manipularlo, la serpiente la mordería; éste es un caso de una percepción errónea. Si la persona percibe correctamente a la serpiente pero decide que no es peligrosa, cuando sí lo es, se tendría una decisión

incorrecta. Finalmente, si la persona percibe correctamente la serpiente como un animal y decide que es peligroso y que por ello debe cerrar el cajón, pero el movimiento es torpe y la serpiente logra atacar entonces se tiene un control de movimiento defectuoso.

Bailey introduce una serie de conceptos para analizar el desempeño de una persona a través de las tres etapas. Tales conceptos son:

- **Habilidad y destreza (Ability and Skill).** “Habilidad” refiere a un rasgo básico de una persona. Muchas de estas habilidades son producto de aprendizajes previos. Estas habilidades se convierten en la base para el aprendizaje de una nueva actividad. Se supone que aquel individuo con muchas habilidades altamente desarrollados pueden más fácilmente ser competente en la realización de una actividad desconocida ([Fle72]). “Destreza” se refiere a un nivel de competencia obtenido en una actividad específica. Cuando se habla de destreza cableando una casa o conduciendo un coche, nos referimos a destrezas específicas.
- **Destreza perceptual (*Perceptual Skill*).** La “Destreza perceptual” consiste en dar coherencia a los datos sensoriales que ingresan a través de los órganos sensoriales y vincular estos datos con el material ya almacenado en la memoria. Los individuos pueden diferir tanto en su habilidad básica de percepción y el nivel de destreza de percepción que han desarrollado. El desarrollo de la destreza de percepción requiere de un ejercicio continuo.
- **Destreza intelectual (*Intellectual Skill*).** Muchas destrezas en las artes probablemente tiene menos que ver con la capacidad de ejecutar respuestas concretas, como hacer un trazo de pincel, que con la decisión de qué colores utilizar en una pintura. Ésta es una destreza intelectual y es análoga a las destrezas de un político o un militar. La “Destreza intelectual” se basa en la vinculación eficiente de la percepción con una acción apropiada a partir del razonamiento, la toma de decisiones y la resolución de problemas.
- **Destreza de control del movimiento (*Movement Control Skills*).** En muchos casos la acción a realizar por los actuadores implica algún tipo de movimiento. Estos se llaman “Destrezas de control de movimiento” e incluyen cosas tales como andar en bicicleta o comer alimentos con un tenedor o con palillos. A medida que estas habilidades se desarrollan, los movimientos se vuelven mas coordinados.

Movimientos bien practicados son conocidos por su falta de control intelectual. Por ejemplo, después de adquirir la destreza de andar en bicicleta, ya no tenemos que pensar el movimiento necesario para mantener el equilibrio.

- **Desarrollo de destrezas (*Developing Skills*).** La adquisición de una destreza no puede aumentar las capacidades básicas, sino que se traduce en una mayor eficiencia de los procesos cognitivos. Los sentidos no parecen mejorar mientras más se utilizan (por ejemplo, el uso constante de los ojos no hace mejorar la capacidad de ver). Sin embargo, el proceso cognitivo sí mejora con el uso. Por ejemplo, para aumentar su fuerza una persona puede incrementar su musculatura a través de ejercicios; estos ejercicios no agregan nuevos músculos sino que modifican los existentes los cuales tienen un límite en su crecimiento. Las destrezas pueden mejorarse sin límite.

4.4.4. Nuestra propuesta para la caracterización del humano

Las caracterizaciones del humano realizadas en el contexto de la Web Semántica se centran principalmente en la información básica sobre el humano, nombre, edad, correo electrónico, etc., mientras que las caracterizaciones en el contexto de HCI se centran en cuestiones fisiológicas. Desde el punto de vista de la visualización ambas propuestas ofrecen información relevante para el proceso de visualización, pero insuficiente. Ninguno de los antecedentes presentados logra capturar las características de la información cultural asociada a la persona.

Es por esto que, para tener en cuenta lo que consideramos esencial en la caracterización del humano, formaremos una propuesta mucho más general que las caracterizaciones existentes. No sólo consideramos importante describir al humano y sus características sino también el modelo cultural que posee. Debe tenerse en cuenta la manera de interpretar la información visual de una persona de oriente no es la misma que la de una persona de occidente. Así mismo, como una visualización se constituye en una interfaz consideramos que es importante también incorporar la descripción del humano desde el punto de vista de HCI. Es por esto que también tomamos algunos de los elementos de la caracterización del humano en HCI para completar nuestro trabajo.

Al describir nuestra caracterización del humano, hablaremos de 3 subcaracterizaciones:

- **Modelo de Humano Básico.** Este modelo busca capturar la información básica de una persona, tal como su nombre, dirección de correo electrónico e información

de localidad. Este modelo se representará utilizando el concepto *Person* de *FOAF*, tal como se presentó anteriormente.

- **Modelo de Procesamiento de información.** Este modelo se basa en el modelo dado en el contexto de HCI dado que una visualización se constituye en una interfaz, por lo que la caracterización dada en HCI es válida para el área de Visualización. Nuestro modelo de procesamiento de información se divide en:

- *Características de los Sensores Externos e Internos.* Aquí se busca medir la capacidad del humano para percibir el mundo. Dos ejemplos de estas características son la capacidad de percibir los colores y la capacidad auditiva. Reconocimiento del Rojo, Verde y Azul: Junto a los problemas de percepción visual que puede llegar a tener una persona también pueden existir problemas en el reconocimiento de colores, no necesariamente ligados a una enfermedad particular. En esta caso nos enfocamos solamente en los tres colores primarios, rojo, verde y azul. Cada uno es medido con un valor entero entre 0 y 100, donde 0 representa la falta de reconocimiento de ese color y 100 el reconocimiento perfecto. Esta información es vital al momento de crear, por ejemplo, mapas de colores.

Capacidad auditiva: La capacidad auditiva es otro de los sentidos que nos proporcionan información sobre el mundo exterior. La utilidad de alarmas o señales sonoras dependerá de la capacidad del humano para escucharlas.

- *Habilidad.* Aquí buscamos representar las habilidades básicas del humano que nos resultan importantes para el proceso de visualización, por ejemplo la memoria visual.

Memoria Visual: Es la capacidad para evocar estímulos visuales presentados previamente. Es la capacidad del humano para asociar lo que ve con lo que vio. Esta capacidad será pesada con valores enteros entre 0 y 100, donde 0 representa la falta de la capacidad y 100 una capacidad perfecta. Al interactuar con una visualización el usuario transita por diferentes vistas, e.g. al hacer *zoom in* se pasa de una visión global a una más particular; la capacidad de asociar la visión particular que se tiene con la visión global que se tuvo depende en gran medida de la memoria visual del usuario. Si el usuario tiene poca memoria visual es necesario reforzar de alguna manera la visión particular con

información que la vincule a la visión global, por ejemplo utilizando una técnica de foco+contexto ([Fur86]).

- *Destreza Perceptual*. Esta característica se centra en la capacidad del humano para interpretar la información percibida por los sensores. Dos ejemplos de esta característica son los problemas de percepción visual y los problemas de percepción de profundidad.

Problemas de Percepción Visual: Existen múltiples enfermedades y desórdenes físicos que afectan la vista de una persona y por lo tanto su percepción visual; miopía, astigmatismo o hipermetropía son algunos ejemplos. Cada uno de estos desórdenes se puede asociar a restricciones o condiciones sobre la visualización, e.g. si el usuario sufre de miopía tiene comprometida su visión lejana, ésta información junto con la distancia entre el usuario y el display permitiría modificar la visualización para que, cuando el usuario se aleje del display, se incluyan objetos visuales más grandes o se utilice una técnica de representación que priorice el *overview*. Esta característica del humano será representada a través de los nombres de las enfermedades que afectan al usuario.

Percepción de Profundidad: Percepción de la profundidad es la capacidad visual de percibir el mundo en tres dimensiones. La percepción de la profundidad permite medir con precisión la distancia desde el que observa hasta un objeto. Todas las técnicas de representación 3D se basan en la percepción de profundidad, al igual que los *displays* 3D. No sería correcto representar una jerarquía de archivos con un árbol en 3D si el usuario no posee o tiene muy poca percepción de profundidad. Esta capacidad se mide con un valor entero entre 0 y 100, donde 0 representa la falta de percepción y 100 una percepción de profundidad perfecta.

- *Destreza Intelectual*. Esta característica busca capturar la destreza del humano para resolver problemas particulares, por ejemplo su destreza en el uso de una computadora y en el uso de una aplicación de visualización.

Destreza con una computadora: Esta destreza se refieren a la capacidad del humano para utilizar el *software* y el *hardware* de una computadora. El conocimiento del nivel de destreza del humano con la computadora es fundamental al determinar cómo el mismo interactuará con el sistema de visualización. No se le puede requerir al usuario que interactúe con

una visualización a través de complejas interacciones si su destreza con la computadora está a nivel *novato*.

Destreza con el sistema de creación de visualizaciones: Esta destreza es similar a la anterior pero focalizada en el uso del sistema de visualización, es decir la aplicación que se utiliza para crear las visualizaciones. El proceso de visualización se constituye en una visualización; la interfaz que el desarrollador ve y las interacciones que utiliza formal la visualización del modelo de visualización. Por lo tanto, la complejidad de la interfaz y de las interacciones va a depender del nivel de expertise del humano. Al igual que en el caso anterior, consideramos tres niveles de habilidad, *novato*, *intermedio* y *experto*.

- *Destreza de Control de Movimiento*. Esta característica busca describir la destreza del humano para realizar movimientos específicos, como tipear o mover el mouse.
 - *Actuadores*. Aquí se busca caracterizar los elementos que forman parte de los actuadores del humano. Por ejemplo, los dedos de la mano y la boca.
- **Modelo Cultural**. Uno de los elementos más necesarios y difíciles de modelar es la relación entre la cultura intrínseca del usuario y las características de una representación visual. Un buen ejemplo de esta relación es la interpretación de los colores; diferentes culturas interpretan el mismo color de diferente modo. El color rojo en la cultura occidental es sinónimo de peligro mientras que en la cultura oriental es sinónimo de buena suerte² (Figura 4.18). Sin duda, estas interpretaciones deben ser tenidas en cuenta al momento de diseñar una visualización. Sin embargo, en la actualidad esto no ocurre. No existen antecedentes en el área de Visualización en donde información del modelo cultural sea usada para definir las características de una visualización. Sí existen antecedentes desde el punto de vista de la Web Semántica e Interfaces de Usuario ([Rei07], [RB09], [RRB07]); en todos estos casos la representación del modelo cultural se basa en el trabajo realizado por Hofstede ([Hof03]) y sus cinco dimensiones culturales (*Power Distance Index*, *Individualism*, *Masculinity*, *Uncertainty Avoidance Index* y *Long-Term Orientation*). Aunque éste es un trabajo significativo en la representación del modelo cultural, en el contexto de Visualización no brinda una contribución relevante. Por este motivo nuestro modelo

²<http://flowingdata.com/2010/04/30/cultural-colors-of-emotion-and-character/>

cultural no se basará en el trabajo de Hofstede sino que será una nueva propuesta orientada a la Visualización.

Por el momento, nuestro modelo cultural sólo abarcará colores. Dicho modelo contendrá mapeos de conceptos a colores, en donde cada mapeo se ubicará dentro de un contexto cultural. Comenzaremos definiendo los conceptos básicos simples tales como *bueno*, *malo* y *neutro*. Además también se incluirán las relaciones entre conceptos: *equivalencia*, *similar* y *opuesto*. Esto nos da la posibilidad de definir nuevos conceptos en función de las relaciones y los conceptos básicos. Por ejemplo, podemos definir el concepto *peligro* como equivalente al concepto *malo*. A partir de esto podemos definir el *Modelo Cultural Occidental* con el siguiente mapeo de conceptos a colores: (*bueno*, *verde*), (*malo*, *rojo*). Podemos extender este modelo incorporando nuevos conceptos como *peligro* = *Equivalente(malo)*.

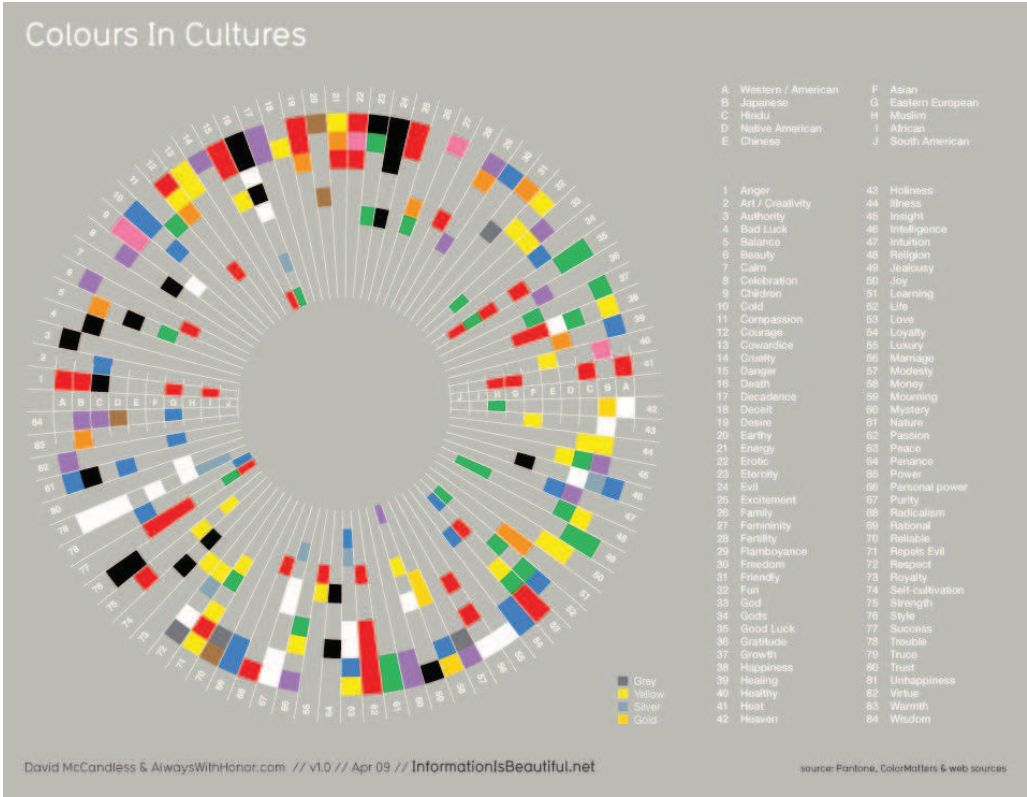


Figura 4.18: Rueda de colores, con sus significados asociados según el modelo cultural.

4.5. Caracterización del proceso de visualización

La visualización es el proceso por el cual se crea una representación visual interactiva a partir de un conjunto de datos. Durante los últimos años fueron presentados diferentes modelos de visualización. Poder caracterizar este proceso nos permitirá definir las ontologías que serán las responsables de representar el proceso de visualización en un contexto semántico.

4.5.1. Antecedentes en la caracterización del proceso de visualización

En el capítulo 2 se hizo un relevamiento de los modelos de visualización presentados en los últimos años. Consideramos que estos modelos constituyen las caracterizaciones del proceso de visualización.

4.5.2. Nuestra propuesta para la caracterización del proceso de visualización

Nuestra propuesta para la caracterización del proceso de visualización será la del “Modelo Unificado de Visualización” ([MCFE03]). Este modelo fue diseñado y desarrollado en nuestro laboratorio y publicado internacionalmente ([MCFE03], [MC06], [UCM06], [ECSM06], [ECM07]). Dicho modelo fue descrito en el capítulo 2.2.2.

Capítulo 5

Ontologías en el proceso de Visualización

En este capítulo se describirá cómo, a partir de las caracterizaciones propuestas, se definieron las ontologías para ser utilizadas como representaciones semánticas en el proceso de visualización. En estas transformaciones de caracterizaciones a ontologías no siempre el mapeo será uno a uno. Ciertamente estas ontologías no representan el estado final de las representaciones y están sujetas a constantes cambios y modificaciones en función de las pruebas y experiencias ganadas al utilizarlas. Las clases, sus relaciones y propiedades serán denominadas usando el idioma inglés para facilitar la reusabilidad de las mismas. En primera instancia se describirá el trabajo realizado por Sebastián Escarza dado que sus resultados son significativos para el desarrollo de esta tesis; posteriormente se presentarán las ontologías desarrolladas para los datos, las tareas, el contexto y el usuario.

5.1. Ontología del proceso de visualización

El trabajo realizado por Sebastián Escarza ([Esc09a], [Esc09b]) constituye la capa axiomática que describe los elementos constituyentes de toda visualización. La ontología se está diseñando siguiendo un enfoque modular por capas que maximiza la integración con tecnologías existentes y la adaptabilidad a los cambios en las mismas. A continuación se describe el trabajo realizado hasta el momento.

5.1.1. Ontología de Visualización

Para obtener una conceptualización compartida y consensuada, el diseño de la ontología se basa en modelos de referencia de visualización preexistentes, principalmente en el MUV ([MCFE03]). En visualización existen muchos trabajos que usan ontologías, pero principalmente se centran en la representación de los datos del dominio del usuario. Muy pocos son los que logran representar en la ontología algún aspecto en particular de la visualización. Estos son casos donde los escenarios son muy acotados y limitados. Estas ontologías, en general, definen conceptos de alto nivel y asocian directamente la semántica de los mismos; en otras palabras, no se deriva semántica a partir de componer otras entidades.

La estrategia utilizada en la tesis doctoral de Escarza es reduccionista (Figura 5.1 y 5.2); la semántica de cada concepto puede ser expresada como combinación de la semántica de cada una de sus partes. Esta es una estrategia *bottom-up* que parte de conceptos primitivos y utiliza constructores que permiten definir ciertos conceptos en término de otros; a estos se los denominan *conceptos derivados*. Dentro de la estrategia utilizada, se busca una arquitectura modular en donde se establece un núcleo ontológico y extensiones que encapsulan definiciones orientadas a propósitos particulares. El núcleo ontológico contendrá los datos del usuario, denominados *Datos Abstractos* en el MUV, el procesamiento de datos, la representación visual, el mapeo visual, el proceso de visualización, los eventos e interacciones y finalmente los conceptos abstractos de la ontología de visualización.

Con respecto a la semántica de los datos, la misma será definida por el usuario, que es quien conoce los datos y su semántica. Dentro del dominio de aplicación se encuentran los datos y los metadatos del usuario. La definición de los tipos de datos se hará utilizando los tipos provistos por OWL, tipos XSD válidos en OWL, y los constructores de OWL para clases y propiedades. Se agregarán categorizaciones según las escalas de los datos: nominal, ordinal, intervalo y razón; tipos de datos no básicos de XSD, como son por ejemplo los números complejos y finalmente se incorporarán estructuras de datos adicionales definidas como clases provistas por la ontología. Un ejemplo de estas categorizaciones se puede encontrar en la descripción de árboles de dialéctica (Figure 5.3 y 5.4) en *Defeasible Logic Programming*¹²

¹<http://cs.uns.edu.ar/~se/OWLTests/delp/DeLP-Example-DeLPOnt.owl>

²<http://cs.uns.edu.ar/~se/OWLTests/delp/DeLP-Example-Individuals.owl>

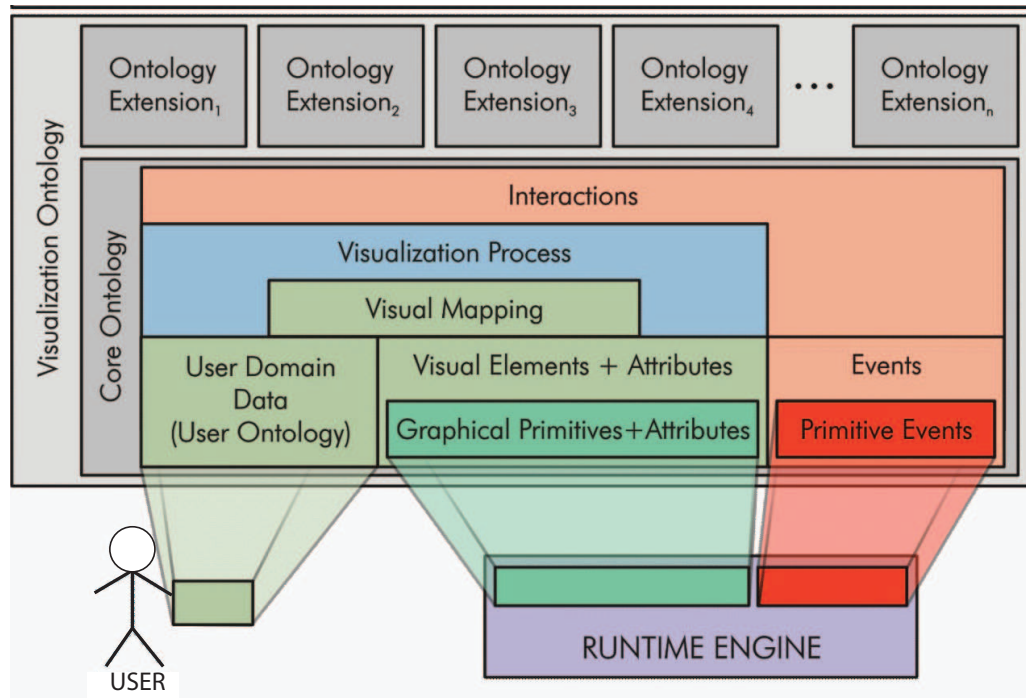


Figura 5.1: Arquitectura de capas en la ontología de visualización.

5.1.2. Procesamiento de datos

Las primeras etapas del proceso de visualización, en general, están más orientadas al procesamiento de datos. En éstas se requiere un mecanismo acorde para realizar este procesamiento sobre las definiciones ontológicas; tal mecanismo debe permitir derivar nuevos datos y metadatos, computar agregaciones, filtrar datos y transformar ciertos datos en otros. Esta última característica permite además definir equivalencias entre definiciones; por ejemplo una rotación puede especificarse como una matriz, una operación con sus parámetros o un quaternion. El esquema de procesamiento de datos se basa en el concepto de *graph patterns*, se definen grafos patrones de entrada, selección y ligadura de variables, como así también grafos patrones de salida para la generación del *output* del procesamiento. Se identifican tres tareas de procesamiento de datos:

1. **Derivación:** Un *input graph* y un *output graph*. Con cada *matching* de *input* se genera el *output*.

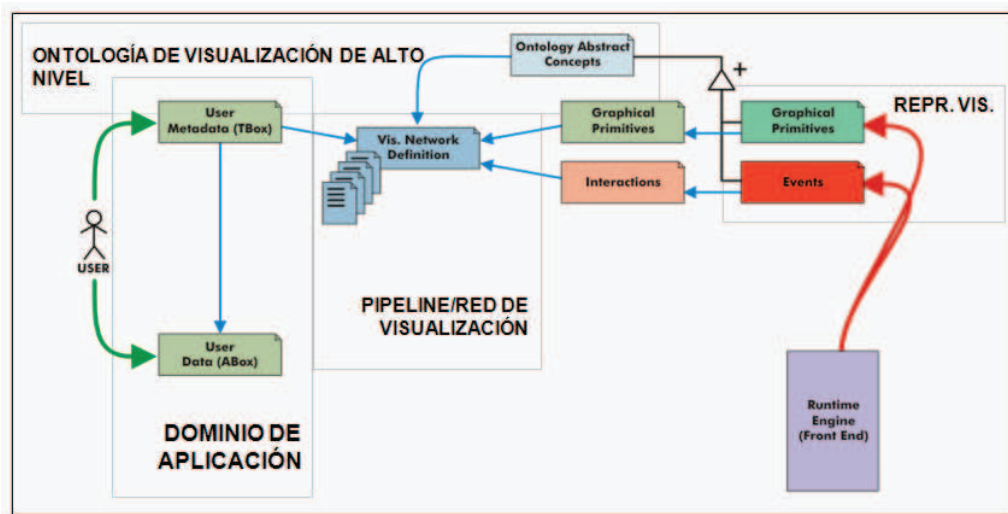


Figura 5.2: Vista de alto nivel de la ontología de visualización.

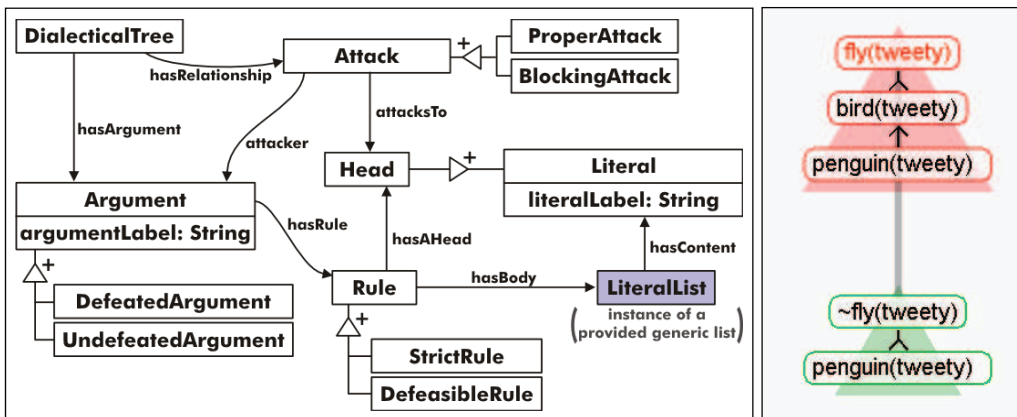


Figura 5.3: Ejemplo de TBox para el caso de árboles de dialéctica en DeLP.

2. **Filtrado:** Un *input graph* y un *output graph*. Con cada *matching* de *input* se genera el *output*. En este caso el *output graph* debería ser un subgrafo del *input graph*.
3. **Agregaciones:** Un *input graph* y un *output graph*. Con cada *matching* de *input*, se acumulan resultados parciales.

El lenguaje utilizado para realizar las consultas sobre los datos es SparQL (Figura 5.5). SparQL es un lenguaje de consulta de RDF, declarativo e inspirado en SQL. Con

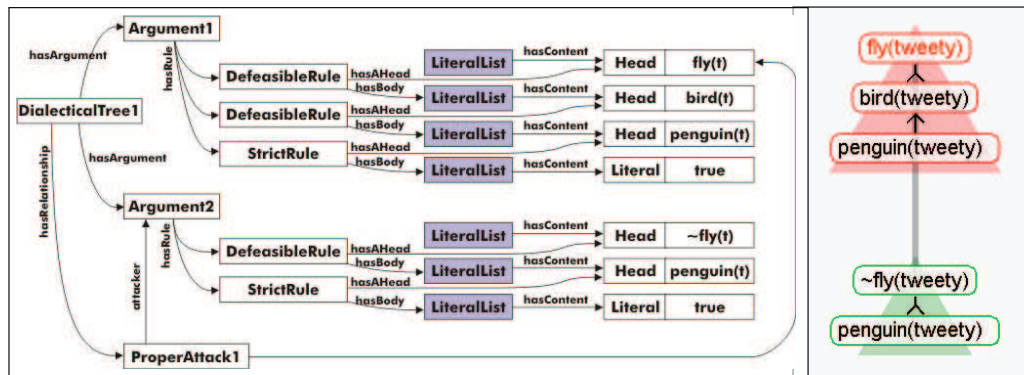


Figura 5.4: Ejemplo de ABox para el caso de árboles de dialéctica en DeLP.

SparQL se pueden definir *graph patterns* para resolver las consultas; una limitación del lenguaje es no permitir consultar caminos.

5.1.3. Representación Visual

Para la representación visual es necesario contar con la definición de tres requerimientos: sustrato espacial, elementos visuales y los atributos de dichos elementos. Los conceptos utilizados para representar el sustrato espacial son:

- **Space:** Espacios matemáticos n-dimensionales.
- **Transformation:** Transformaciones matemáticas generales entre espacios. Se representan como la transformación, pudiendo ser ésta una matriz, expresiones, etc. Cada transformación contiene un volumen de *clipping* en origen y podrían tener un volumen de destino, la inclusión de este último aún no está definida.

Un algoritmo de diagramado simplemente calcula transformaciones para posicionar los elementos visuales en el espacio. Las transformaciones no primitivas se definen y son transformadas a su vez en transformaciones primitivas para ser ejecutadas. Los elementos visuales constituyen el contenido de los espacios. Estos pueden ser primitivas gráficas provistas por el motor de ejecución o elementos derivados mediante transformaciones y subespacios. Los elementos visuales, así como las transformaciones, dependen directamente de las primitivas soportadas por el motor de ejecución. En su tesis doctoral Escarza ([Esc09a], [Esc09b]) plantea un motor minimal. Actualmente, los

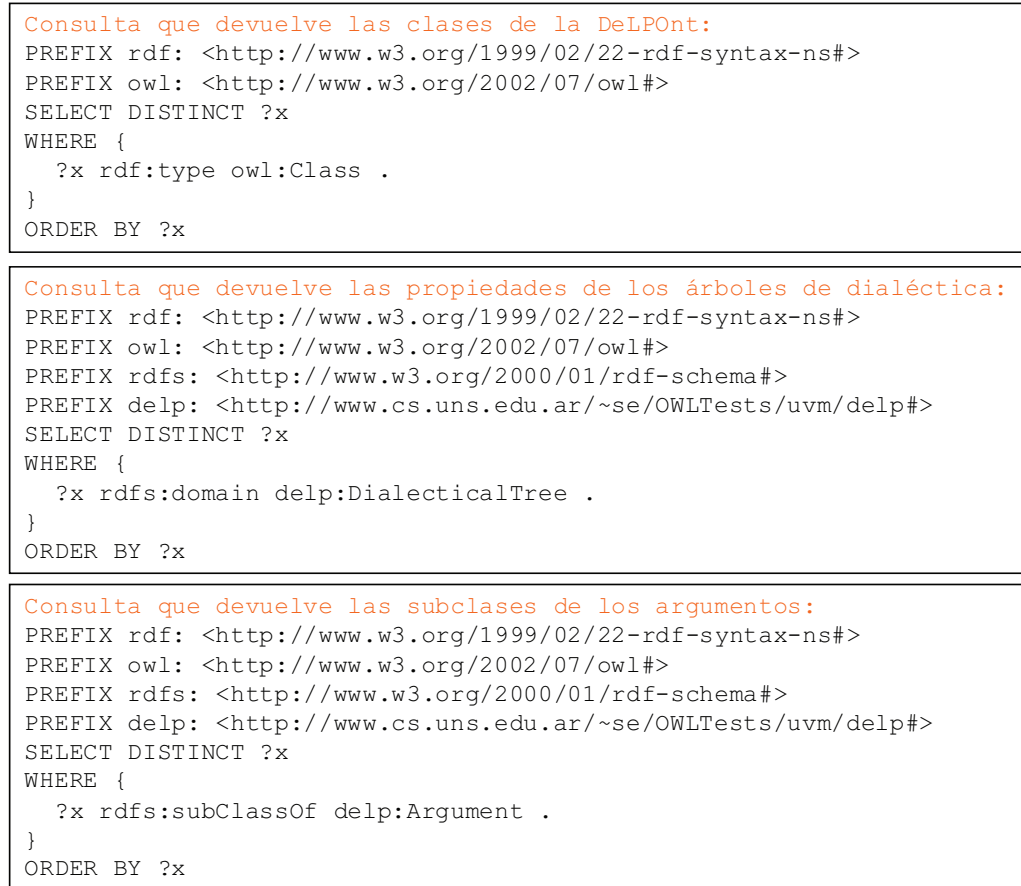


Figura 5.5: Ejemplo de consultas en SparQL.

elementos visuales se enfocaran en primitivas basadas en vértices, para lo cual se definió el concepto abstracto de *vertex*.

Los atributos de los elementos visuales son propiedades de las ontologías que ligan las entidades a los valores de los mismos. Inicialmente se consideró acotar el problema, considerando únicamente la representación de materiales como atributo gráfico de una primitiva; esto incluye color, modelos de iluminación, sombreado, etc. Los atributos visuales se asocian a las primitivas gráficas y no a los elementos visuales derivados.

Los espacios de color se consideran espacios, instancias de *Space*. Las primitivas pertenecientes a los mismos son *Color Values*, que son vértices asociados a espacios de color. Todas las operaciones sobre espacios geométricos convencionales están disponibles y es posible utilizar *aliases* para nombrar colores, por ejemplo azul, índigo o Pantone 536-1

CVS.

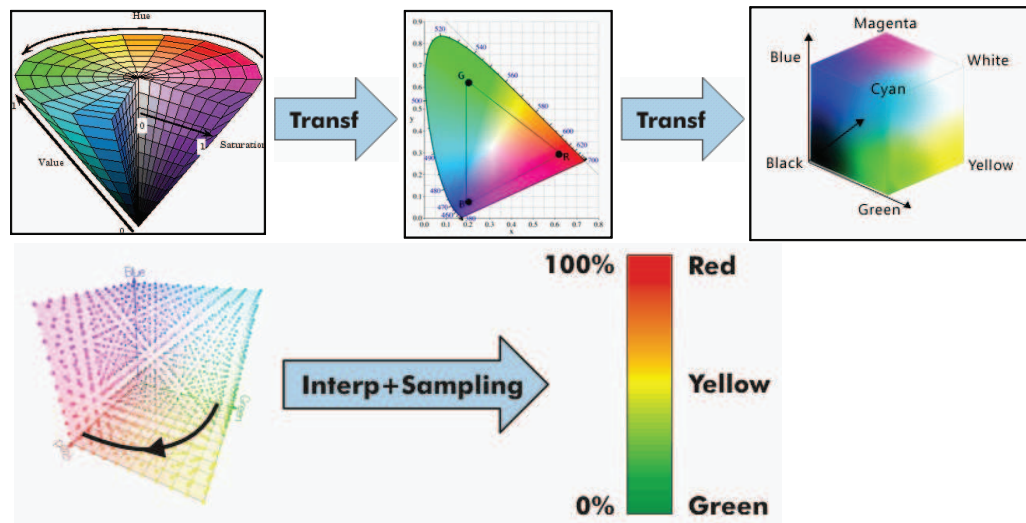


Figura 5.6: Los espacios de color se representan como una instancia del concepto *Space*.

El mapeo visual se define teniendo en cuenta dos aspectos: las transformaciones de datos y el proceso de visualización. Las transformaciones de datos establecen mapeos de entidades abstractas en elementos visuales, también establecen mapeos de atributos de los datos en atributos gráficos y definen propiedades ontológicas para hacer los *bindings*. El proceso de visualización no siempre puede representarse como una asociación de pares (*dato, entidad visual*) ya que muchas veces es resultado de complejas transformaciones. Dicho proceso puede verse como una traducción de los datos a un lenguaje visual, tal como lo describió Mackinlay ([Mac86]).

Las entidades genéricas son provistas por los conceptos abstractos de la Ontología. Por ejemplo, para la familia de técnicas sobre árboles, se proveerá la estructura conceptual necesaria para describir árboles. De esta manera, una técnica sobre árboles no se encuentra definida en términos de los metadatos del usuario para el dominio particular visualizado, alcanzando así una aplicabilidad más general.

5.1.4. Proceso de visualización

El proceso de visualización, también denominado “Red de Visualización” requiere la definición de dos elementos, las etapas y los *streams*, representando estos últimos las

conexiones entre las etapas. Actualmente se está analizando si se incorporará la ontología de proceso de OWL-S para describir el comportamiento de cada etapa y si el modelo de proceso de datos basado en *graph patterns* tiene suficiente poder expresivo para representar los procesos existentes en visualización.

El motor de ejecución publica los eventos que es capaz de manejar (Figura 5.7). En base a dichos eventos primitivos se pueden definir eventos derivados, análogamente a lo que ocurría con las primitivas gráficas y los elementos visuales. Paralelamente, la ontología o el usuario, define las interacciones posibles. Cada interacción actúa sobre la Red de Visualización modificando su comportamiento. Mediante asociaciones se determina qué conjunto de eventos disparan qué interacción.

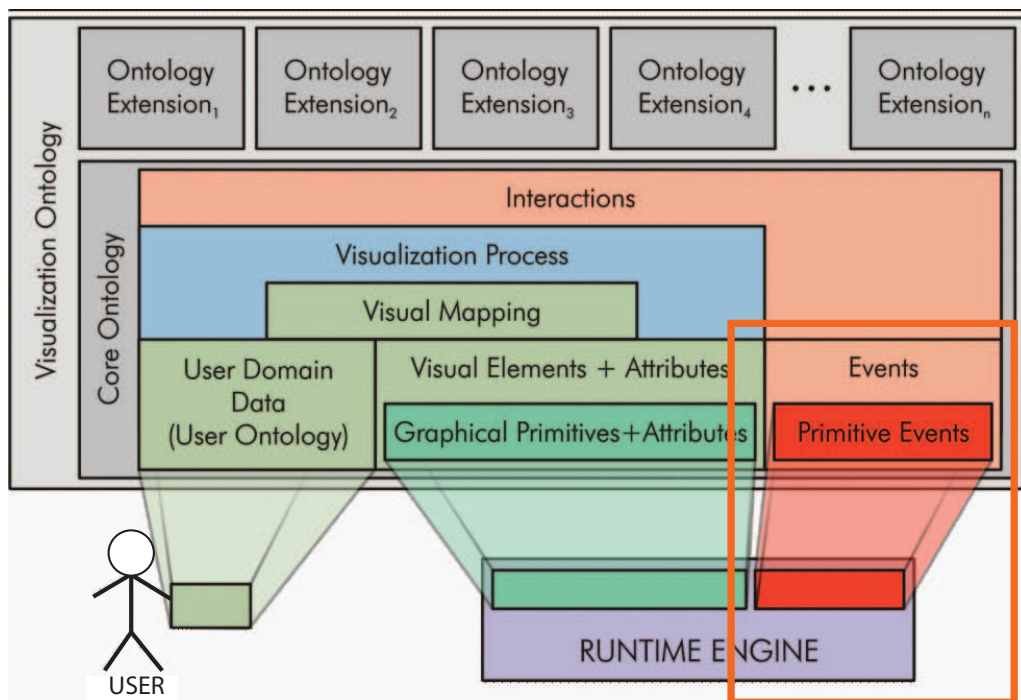


Figura 5.7: Manejo de eventos en la ontología de visualización.

5.1.5. Extensiones

Las extensiones de la ontología cubren aspectos no básicos, tienen propósitos particulares y son independientes entre sí. Ejemplos de estas extensiones son: la gestión

de color y la taxonomía de formas. A continuación detallaremos nuestra propuesta para una ontología de color, que se basa en la representación de espacios vista anteriormente ([Esc09a], [Esc09b]).

Ontología de Color

En nuestra propuesta inicial representaremos solamente el espacio de color RGB (*red*, *green*, *blue*). La ontología de color tendrá un solo concepto denominado *Color* el cual tendrá como superclase a la clase *Vertex* ([Esc09a]). La representación del concepto de color es relativamente sencilla; sólo utilizamos los atributos *x*, *y*, *z* heredados de *Vertex* para indicar los valores de rojo, verde y azul respectivamente. En esta ontología no sólo representaremos el concepto de color sino también sus relaciones con otros colores. Dentro de un espacio de colores, es posible establecer relaciones entre colores. Por ejemplo, utilizando como base una rueda de colores discretizados (Figura 5.8) podemos establecer la relación *opuesto* donde dado un color cualquiera su *opuesto* es el color que se encuentra en el otro extremo de la rueda.

La importancia de poder representar este tipo de relaciones radica en que muchos mapeos visuales se pueden beneficiar de esta información semántica. Supongamos que gracias a la información semántica de los datos de entrada sabemos que dos datos representan información opuesta; al momento de seleccionar los colores para representarlos sería correcto que dicha relación se mantenga en los colores a utilizar. Por el momento nuestra ontología incluirá dos relaciones, es decir, dos *object properties* con rango y dominio en la clase *Color*. Una de ellas es *opposite* y la otra *next*. La *object property opposite* vinculará un color con su opuesto de acuerdo al espacio de colores discretizado mientras que *next* relacionará a un color con su siguiente en el espacio de colores discretizado (Figura 5.9). Es importante destacar que extender esta ontología para agregar nuevas relaciones es una tarea muy sencilla.

En la ontología de color se está describiendo cómo se constituye un color y sus relaciones con otros colores, sin embargo esta ontología no contiene colores *per se*. Para incorporar las instancias del concepto *Color* y sus relaciones se decidió incluir una segunda ontología, la cual importa la ontología de color. Esta nueva ontología se denominó *ontología de colores*. La decisión de separar el concepto de color de las instancias de colores se basó en la flexibilidad que ofrece esta solución para poder cambiar fácilmente el conjunto de colores que manejará el sistema.

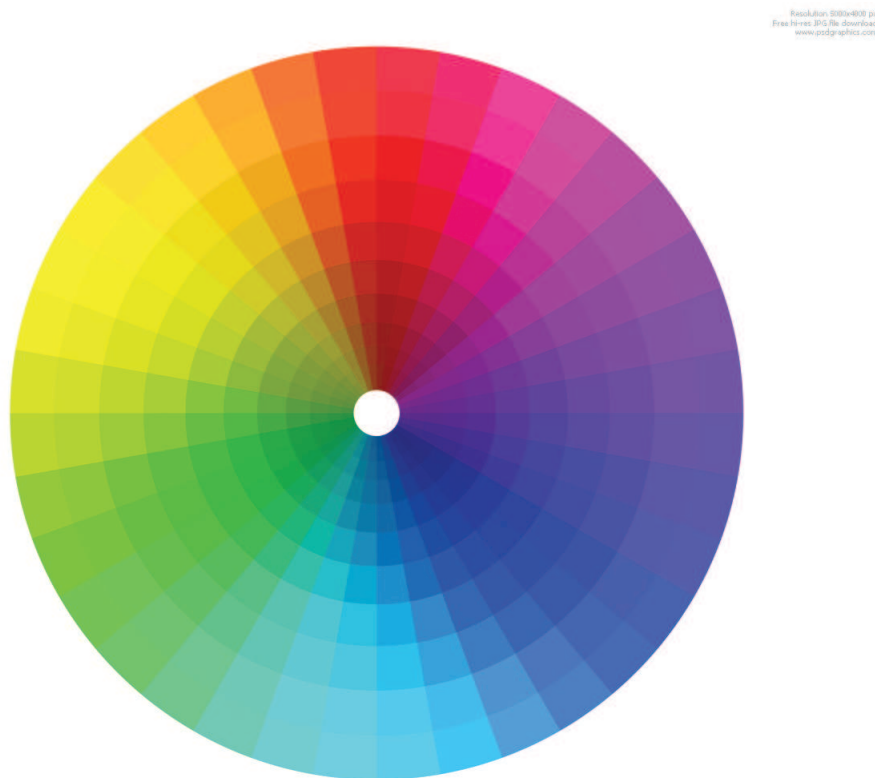


Figura 5.8: Utilizando el espacio de colores discretizado es posible establecer relaciones entre los colores.

Ontología de Colores

La ontología de colores tendrá las instancias del concepto *Color*, cada una de ellas con sus respectivas relaciones de *opposite* y *next*. Con esta ontología, al seleccionar un color específico, se tiene conocimiento de los restantes colores relacionados con él.

5.1.6. Ontología de visualización y visualización basada en semántica

El aporte principal de esta tesis lo constituye la definición de la capa denominada *capa de inferencia* que contiene las reglas de inferencia correspondientes (Figura 5.10). Dichas reglas se basarán en las ontologías descritas en este capítulo y en el uso de un razonador semántico, tal como se describirá en el próximo capítulo.

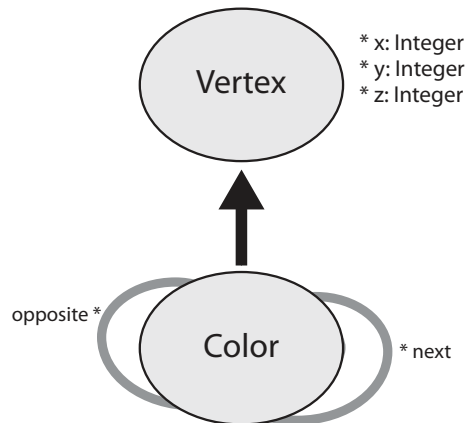


Figura 5.9: Ontología de Color.

5.2. Ontología de los datos

Para definir la ontología de los datos se sigue la propuesta realizada en el capítulo anterior. Se comienza definiendo el concepto de dato denominado *Data* el cual se representa como una clase. *Data* contiene solamente un *data property* de tipo *string*, el contenido de esta propiedad se corresponde con el nombre del dato que se está representando. La Figura 5.11 resume el contenido de la ontología. En el contexto del ejemplo presentado del hospital (Sección 4.1.5) para representar el dato *Paciente* tendríamos una instancia de *Data* cuyo atributo *name* sería “Paciente”.

A continuación se incluyen los conceptos de dato atómico y compuesto con las clases *Atomic* y *Composite* respectivamente. Ambas clases tienen como superclase a *Data*. De la clase *Atomic* se derivan dos nuevas clases, *Discrete* y *Continuous* para representar los conceptos de datos discretos y continuos. Cada tipo propuesto se representa con una clase; como resultado se incorporan 5 nuevas clases, *Ordinal* y *Nominal* cuya superclase es *Discrete* y *Ratio*, *Interval* y *Absolute*, con superclase *Continuous*. Para representar los tipos de datos compuestos se incluyen 4 clases, correspondientes con los tipos presentados en el capítulo 4, *Set*, *List*, *Tree* y *Graph*. Para los tipos de datos compuestos es necesario describir las características del tipo de dato que lo compone. Si el dato compuesto es homogéneo entonces solo es necesario dar una única descripción del dato contenido; sin embargo, si el dato compuesto es heterogéneo es posible tener que dar una descripción para cada dato incluido en la estructura. Para modelar esta situación se crearon las clases *Homogeneous* y *Heterogeneous*. La clase *Homogeneous* posee una *object property*

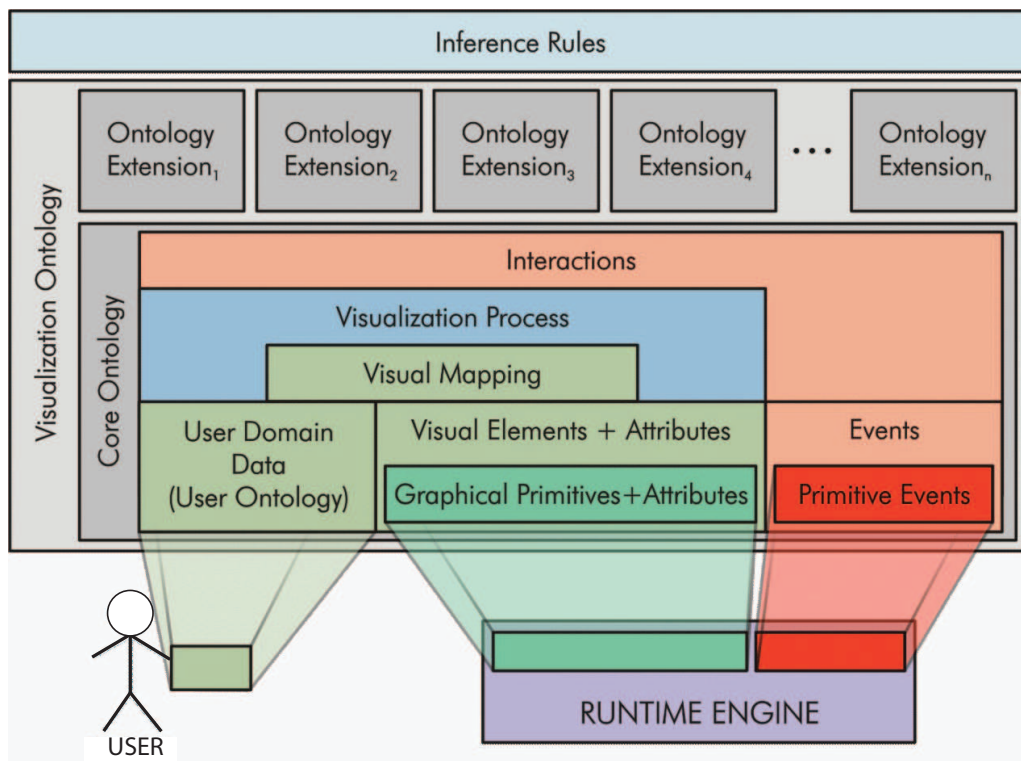


Figura 5.10: Diagramado de la ubicación de la capa de inferencia.

functional que la relaciona con la clase *Data*, denominada *hocontains*. De esta forma estamos indicando que una estructura homogénea tiene asociada la descripción de uno y sólo un dato, que corresponde a la descripción de la estructura de todos los datos que la componen. Para el caso de *Heterogeneous* se incluye una *object property* llamada *hecontains* que asocia la clase *Heterogeneous* con *Data* pero esta relación es del tipo 1-a-muchos. A través de esta relación indicamos que, si una estructura es heterogénea entonces para describir la estructura de sus datos es necesario describir las estructuras asociadas a cada tipo de dato incluido. Para poder conectar los tipos de datos compuestos con las clases *Homogeneous* y *Heterogeneous* se creó una clase *Composition* que constituye una superclase para *Homogeneous* y *Heterogeneous* y finalmente se incluyó una *object property functional* en *Composite* denominada *content* que tiene como rango a la clase *Composition*.

Para representar nuestro ejemplo de “Paciente” tendríamos una instancia del concepto *Set* cuyo *data property name* sería “Paciente”. Esta instancia se relacionaría con una

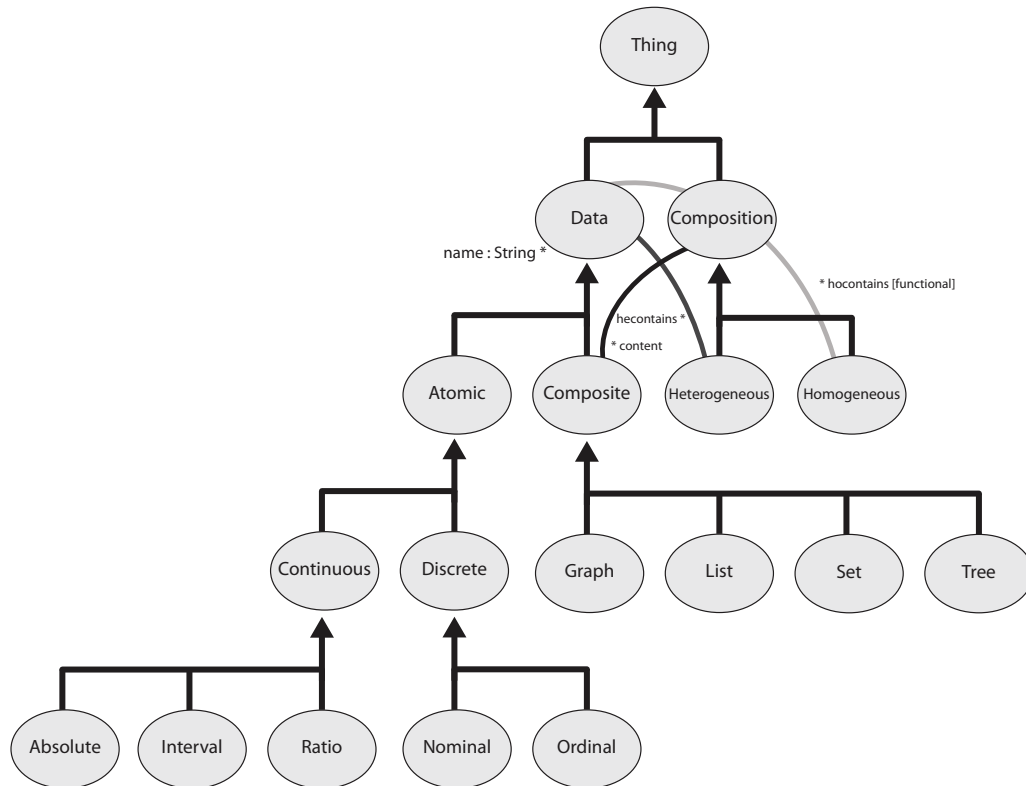


Figura 5.11: Diagrama de clases y relaciones para la ontología de datos propuesta.

instancia del concepto *Heterogeneous* que a su vez se relacionaría con otras cuatro instancias, una para cada tipo de dato en “Paciente”. Específicamente tendríamos instancias de los conceptos *Nominal*, *Ordinal*, *Absolute* y *Nominal* y cada una de estas instancias tendría asociado a su atributo *name* los *strings* “Nombre”, “ID”, “Edad” y “Género”, consistente con el ejemplo dado en el capítulo anterior (Figura 5.12).

5.3. Ontología de las Tareas

Tal como se describió en el capítulo anterior, nuestra caracterización de tareas incluye nueve de ellas: *Identificación*, *localización*, *diferenciación*, *agrupación*, *ranking*, *comparación*, *asociación*, *correlación* y *exploración*. Esta ontología busca describir las características de cada tarea de forma tal que se puedan asociar dichas características a elementos del mapeo visual. La ontología *Tasks* contendrá un concepto para cada tarea, es decir tendremos nueve conceptos: *Identification*, *Location*, *Differentiation*,

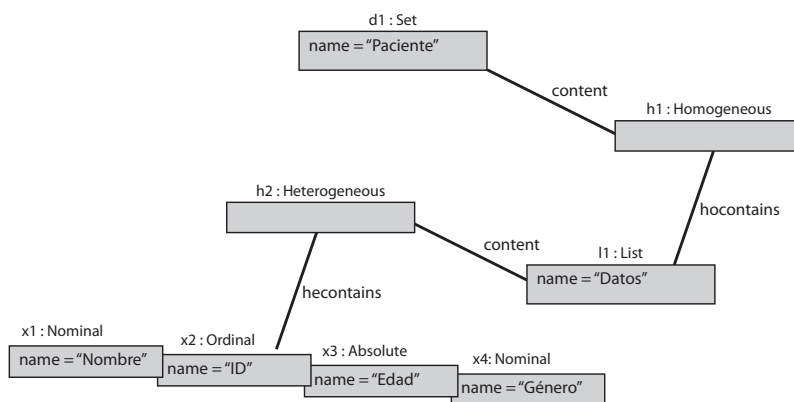


Figura 5.12: Representación del dato “Paciente” con nuestra ontología de datos.

Grouping, Categorization, Ranking, Comparison, Association, Correlation, Exploration (Figura 5.13).

La identificación, concepto *Identification*, busca detectar uno o más elementos dentro de un conjunto de ellos. Esto implica que, al hablar de identificación, tendremos los conceptos de elemento detectado y elemento descartado. El elemento detectado es aquél que cumple con las condiciones de la identificación mientras que el elemento descartado es aquél que no cumplió con tales condiciones. Por esto incluiremos dos conceptos, *Detected* y *Omitted*. Crearemos dos *object property functional*, *detectedElement* y *omittedElement* que tendrán como dominio a *Identification* y rango *Detected* y *Omitted* respectivamente. El objetivo de describir una tarea de esta manera es que nos permitirá, en caso de que el usuario quiera realizar una identificación, asociar al concepto *Detected* con un concepto visual. Por ejemplo, si se asocia el concepto *Detected* con el color rojo entonces aquellos elementos detectados por la tarea *Identification* serán pintados de rojo.

La localización, concepto *Location*, busca establecer una región espacial a partir de cierta condición de locación. Para representar esto utilizaremos un concepto *SpatialRegion* que será luego asociado con diferentes representaciones de regiones, ya sea 2D o 3D, con colores con transparencia o sin ella. El concepto *Location* será el dominio de una *object property functional use* que tendrá como rango a el concepto *SpatialRegion*.

La diferenciación, concepto *Differentiation*, busca distinguir elementos como distintos. Para esta tarea consideraremos la posibilidad de que la diferenciación dé como resultado más de 2 elementos o grupos de elementos. Es decir, necesitaremos de más de 2 formas de diferenciar. En este caso tendremos el concepto de *Differentiated* el cual se asociará con

una forma específica de diferenciar. El vínculo entre ambos conceptos se hará con la *object property contains* que tendrá como dominio a *Differentiation* y rango *Differentiated*.

La agrupación y la categorización, conceptos *Grouping* y *Categorization*, presentan en su composición la misma forma. Ambas tareas tratan con grupos; la diferencia radica en cómo se definen dichos grupos. Aunque a nivel de conceptos se presentaran por separado y la semántica es distinta, la estructura de ambas será igual. Cada concepto tendrá asociado un conjunto de grupos los cuales, a su vez, serán asociados con características visuales. De estos se desprende que es necesario contar con el concepto de *Group* y una *object property includes* cuyo dominio será *Grouping* y *Categorization* y rango *Group*.

La exploración, concepto *Exploration*, busca permitir al usuario la libertad de “recorrer” la visualización. Esta tarea no impone un nuevo mapeo visual sobre los datos. Recordemos que al crearse la visualización ya se le dio a cada dato a visualizar un mapeo visual. Este concepto no tendrá otros conceptos asociados.

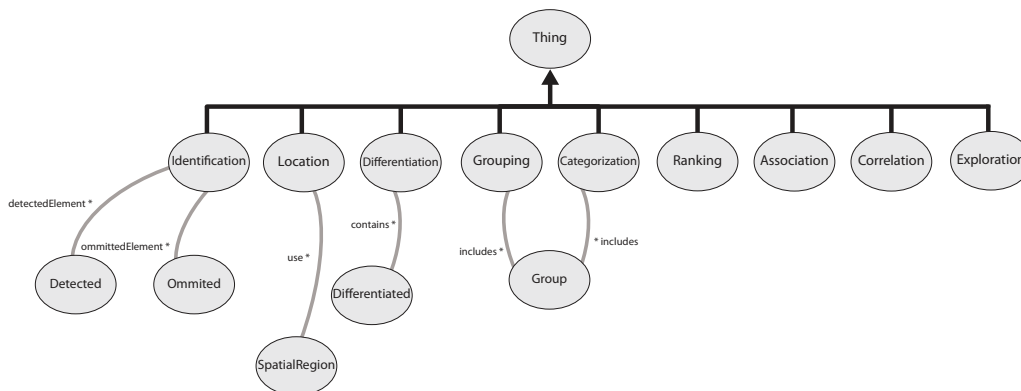


Figura 5.13: Ontología de Tareas.

5.4. Ontología del Contexto

Como se presentó en el capítulo anterior, el contexto incluye el *hardware* de base en donde se ejecuta el proceso de visualización, el ambiente físico en el que el usuario se encuentra y los dominios organizacionales y de aplicación. Esta ontología, denominada *Contexto*, se formará integrando cuatro ontologías, la ontología *Hardware*, la ontología *Ambiente Físico*, la ontología *Dominio de Aplicación* y la ontología *Dominio Organizacional*.

5.4.1. Ontología del *Hardware*

La ontología *Hardware* contiene la semántica necesaria para describir los componentes duros de la computadora donde se esté realizando la visualización. Las cinco categorías presentadas anteriormente fueron transformadas en clases (Figura 5.14); éstas son respectivamente las clases *Input*, *Output*, *GPU*, *CPU* y *Other*. Las clases *Input*, *Output* y *Other* funcionan como agrupaciones de otras clases mientras que *CPU* y *GPU* son descripciones de los dispositivos propiamente. Se creó una clase *Device* que constituye la clase general para todos los dispositivos de entrada y salida. Esta clase contiene los *data properties* *brand*, *model* y *transfer rate*; este último tiene como rango el tipo *double* y los demás son *string*. Las clases que se describen tanto para *Input* como para *Output* extienden la clase *Device* (Figura 5.14).

Clase *Input*

La clase *Input* contiene la semántica asociada a los dispositivos de entrada. Se crearon cinco clases correspondientes a *Mouse*, *Keyboard*, *Joystick*, *Glove* y *TouchSurface*. Estas clases se relacionan a través de *object properties* con la clase *Input*. Para simplificar la notación, todo *object property* se denominó *relatedTo* seguido del nombre de la clase con la que se relaciona. Es decir, la clase *Input* contiene cuatro *object properties*: *relatedToMouse*, *relatedToKeyboard*, *relatedToJoystick* y *relatedToGlove*. Cada una de estas propiedades tiene como dominio la clase *Input* y rango en la clase destino.

La clase *Mouse* contiene un *data property* llamado *dimension* y tiene como rango el tipo *integer*. Esta propiedad indica el número de dimensiones soportadas por el mouse, permitiendo así dar soporte a mouse 2D y 3D. Se incluye también una propiedad de tipo *integer* para denotar la cantidad de botones presentes, tal propiedad se llama *buttonsCount*. En el contexto de los botones de un mouse es importante destacar si el mouse tiene soporte para botones analógicos y/o modales; para esto se incluyen dos propiedades de tipo *boolean* denominadas *supportAnalogButtons* y *supportModalButtons*.

La clase *Keyboard* contiene un *data property* denominado *type* el cual es de tipo *string* e indica el tipo de teclado con el que se cuenta. Esta propiedad sirve para identificar si se cuenta con un teclado tipo QWERTY o uno reducido, como puede ser el de un celular o netbook. También se tiene una *data property* llamada *size* que indica el tamaño promedio de las teclas del teclado. Esta propiedad es importante ya que, si se cuenta con

teclas pequeñas, se debe evitar agrupar comandos en teclas muy próximas debido que podrá resultar dificultoso para el usuario utilizarlas simultáneamente.

La clase *Joystick* tiene también las *data properties* *dimension* y *buttonsCount*, al igual que la clase *Mouse*, con los mismos objetivos. También se incluyen las propiedades de tipo *boolean* *supportAnalogButtons* y *supportModalButtons*. La última clase que se incorporó, en el contexto de los dispositivos de entrada, es la clase *Glove*. Los guantes de realidad virtual resultan muy efectivos al utilizarlos con interacciones 3D, en especial si se cuenta con un monitor 3D.

La clase *TouchSurface* tiene un *data property* de tipo booleano denominado *supportMutitouch* el cual indica si la superficie es *multitouch*. Por el momento la clase *Glove* no incluye propiedades debido a que aún no se ha definido qué elementos se tendrán en cuenta para describir un guante virtual.

Clase *Output*

La clase *Output* contiene la semántica asociada a los dispositivos de salida. Se crearon tres clases correspondientes a *Monitor*, *HeadMountedDisplay* y *Projector*. Estas clases se relacionan a través de *object properties* con la clase *Output*. Al igual que con la clase *Input*, todo *object property* se denominó *relatedTo* seguido del nombre de la clase con la que se relaciona. Para las tres clases mencionadas se incorporó un *data property* *supportColor* de tipo booleano para indica si el dispositivo de salida soporta colores. También se considera como parte de la salida un sistema de sonido pero por el momento esto sólo se representa como un *data property* de tipo booleano *existSoundSystem*, que indica si se cuenta o no con sonido.

La clase *Monitor* tiene la *data property* *dimensions* que se utiliza para indicar si el monitor es 2D o 3D. También existen propiedades relacionadas con el tamaño físico de la pantalla y la máxima resolución permitida, *size* y *resolution* respectivamente; ambas son *data properties* de tipo entero.

Clase *GPU*

La clase *GPU* contiene un *object property* de tipo *CPU* denominado *relatedToCPU*. La función de esta propiedad es la de describir el tipo de unidad de procesamiento que tiene el

sistema gráfico. Además, esta clase contiene una *data property* para indica la cantidad de memoria presente en el sistema gráfico, dicha propiedad se llama *memorySize*. Se incluyen dos *data properties* booleanos, *directXSupport* y *openGLSupport*, para indicar si el sistema gráfico soporta DirectX y OpenGL. Además se tienen otros dos *data properties* de tipo *double* para indicar la versión de API soportada, *directXVersion* y *openGLVersion*.

Clase *CPU*

Esta clase extiende a la clase *Device* presentada anteriormente. La clase *CPU* contiene dos *data properties*: *Count* y *Speed*, la primera con rango *integer* y la segunda con rango *double*. La propiedad *Count* indica la cantidad de procesadores presentes y *Speed* la velocidad de procesamiento. Para indicar el tipo de cpu se utilizan los atributos *brand* y *model* de la clase *Device*. Por el momento se está considerando solamente arquitecturas homogéneas.

Clase *Other*

La clase *Other* es utilizada para representar la semántica de aquellos elementos no considerados en las clases anteriores. Por el momento sólo se almacena información relacionada a la memoria de la computadora. La cantidad de memoria se representa con la *data property* *memorySize*, mientras que la velocidad se representa con la *data property* *speed*.

5.4.2. Ontología del Ambiente Físico

La ontología *Ambiente* contiene una descripción semántica de las características físicas del ambiente en donde se desarrolla la visualización. Para esta ontología crearemos las clases *Light*, *Noise*, *Temperature* y *UserDevice*. La clase *Light* contiene dos *data properties* de tipo *double*, *intensity* y *reflection*. La primera indica la intensidad de la luz en el ambiente y la segunda indica la intensidad de luz directamente aplicada sobre el *display*. La clase *Noise* tiene tres *data properties* de tipo *double*, *continuous*, *intermittent* e *interference*. La clase *Temperature* sólo tiene un atributo de tipo *double* que es *degrees*. La temperatura se mide en grados Celsius. Finalmente la clase *UserDevice* se utiliza, por el momento, solamente para indicar la distancia entre el usuario y el *display* a través de la *data property* *distance* de tipo *double* (Figura 5.15).

5.4.3. Ontologías del Dominio de Aplicación y Dominio Organizacional

Estas ontologías, al igual que la ontología *Modelo Cultural*, asocian conceptos con características visuales. Por ejemplo, en la ontología *Datos* podría incluirse el tipo de dato *agua*; este concepto tendría asociada la descripción del tipo de dato, en el contexto de la ontología *Datos*, y además estar asociado, en el contexto de la ontología *Dominio de Aplicación* con el concepto *verde* de la ontología *Colores*. Sin duda este tipo de representación de conocimiento es un área muy amplia y todavía no se ha resuelto cuál es la mejor forma de implementar una ontología para modelar este problema.

5.5. Ontología del Usuario

Tal como se indicó en el capítulo 4.4, para la representación de la ontología del usuario se utilizará el trabajo realizado en FOAF. Sin embargo, el concepto de persona utilizado en esta ontología, *Person*, no alcanza para cubrir la semántica necesaria a considerar dentro del proceso de visualización. Los dos elementos principales que quedan por fuera de FOAF son, el modelo cultural y el modelo de procesamiento de información. En el contexto de la ontología del usuario, se creará un concepto *User* que tendrá una relación *is-a* con el concepto *Person* de FOAF. Este nuevo concepto servirá como dominio para las relaciones con el modelo cultural, el modelo de dominio y el modelo de procesamiento de información.

5.5.1. Ontología del Modelo Cultural

Esta ontología se representará de igual forma que las ontologías *Dominio de Aplicación* y *Dominio Organizacional*. Al igual que en estos dos casos, todavía no se ha resuelto cuál es la mejor forma de implementarla.

5.5.2. Ontología del Modelo de Procesamiento de Información

Esta ontología contendrá seis conceptos, los cuales se corresponderán con lo visto en el punto 4.4.4. El primer concepto de esta ontología será el concepto *Sensors*, el

cual contendrá información asociada a los sensores externos e internos. Tendrá dos *data properties* de tipo *Integer*, *ColorRecognition* y *HearingAbility*. La primera se asocia al elemento "Reconocimiento de los Colores Rojo, Verde y Azul" y la siguiente a "Capacidad Auditiva". El siguiente concepto es *Abilities*, por el momento contendrá un solo *data property* de tipo *Integer* denominado *VisualMemory*. A continuación se incluirán los conceptos *PerceptualSkills* y *IntellectualSkills*. En el primero se tendrán dos *data properties* de tipo *Integer*, *VisualPerception* y *DepthPerception*. El concepto *IntellectualSkills* contendrá dos *data properties* de tipo *String*, *ComputerSkills* y *VisualizationApplicationSkills*, los valores presentados en 4.4.4 para las características "Destreza con la Computadora" y "Destreza con el Sistema de Visualización" se representarán como valores de tipo *String*. Los últimos dos conceptos serán *MovementControlSkills* y *Responders*.

5.6. Ontología de Valoración

En la visualización basada en semántica, una pieza de semántica fundamental es un juicio de valores sobre los datos, las técnicas y los elementos involucrados. En otras palabras, si se tiene un conjunto de datos que varían entre 0 y 100 y a partir de la semántica el usuario describe que mientras más cerca de 100 se esté es más peligroso, esta última descripción representa un juicio de valor sobre el dato. Esto es fundamental para poder hacer una mejor visualización, nosotros sabemos que algo peligroso es malo, por lo que mientras más cerca de 100 esté el dato, peor será. En nuestra percepción lo malo se asocia con el color rojo por lo que podemos inferir que los datos cercanos a 100 se mostrarán en un rango de rojo. Describiremos entonces cómo el usuario puede asociar juicios de valor a los datos. La inferencia será responsabilidad del razonador semántico.

Por el momento nos concentraremos en la valoración de los datos a visualizar; esto significa que la valoración será realizada sólo sobre los tipos de datos definidos. La ontología de valoración contará con tres conceptos iniciales *good*, *bad* y *neutral*; además se contará con la posibilidad de definir dos relaciones *equivalent* y *opposite*. Los tres conceptos definidos también se incluirán en las ontologías *Modelo Cultural*, *Dominio de Aplicación* y *Dominio Organizacional*. Para cada tipo de dato se podrá definir una cota superior e inferior y la valoración que se asocia a cada una.

Supongamos que se quiere representar la temperatura del agua durante un experimento químico. En este contexto, una temperatura por arriba de 100 grados será considerada peligrosa. En nuestro modelo cultural asociaremos a *bad* el color *rojo* y a *agua* el color *azul*. En la ontología *Valoración* incluiremos el concepto *peligro* como equivalente a *bad* y en la ontología *Datos* diremos que *temperatura* es de tipo *razón*. Para poder visualizar la temperatura será necesario crear un mapa de colores y dicho mapa será continuo ya que el tipo de dato es razón; sabemos que *peligro* es equivalente a *bad*, el cual se presenta con *rojo*. Se deduce de esto que *peligro* se representará con *rojo*, como *peligro* se asoció a *temperatura* en la cota superior 100, en el mapa de colores el valor 100 corresponderá a *rojo* y como el mapa es continuo se generará una degradación de *rojo* a *azul* para el resto del mapa. Queda aún por definir qué otros elementos son necesarios conocer para poder completar el proceso, por ejemplo ¿se puede crear un mapa de colores conociendo la cota superior de un dato pero no la inferior? ¿qué color se usará cuando la temperatura supere los 100 grados?

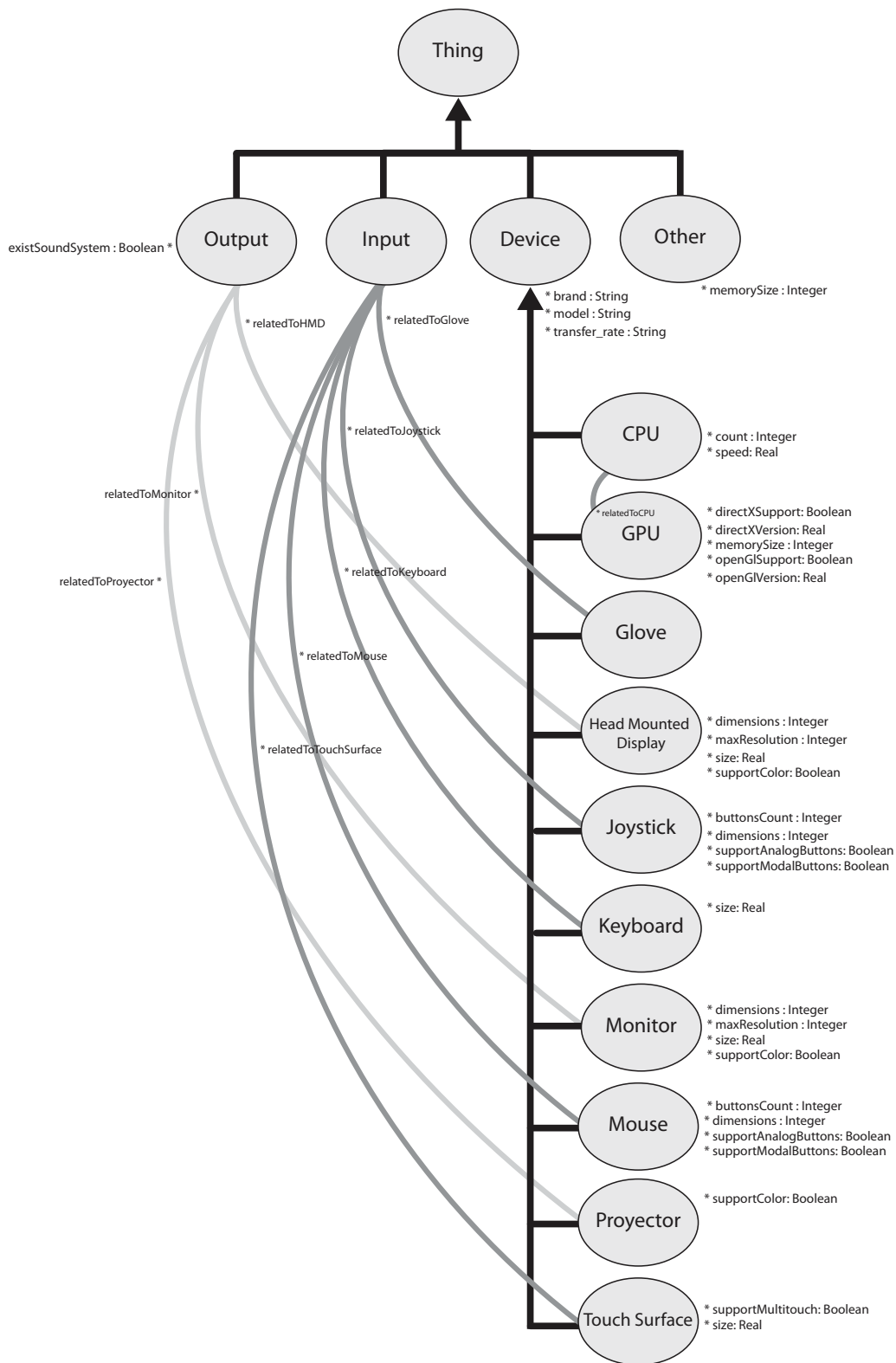


Figura 5.14: Diagrama de clases para la ontología de hardware propuesta.

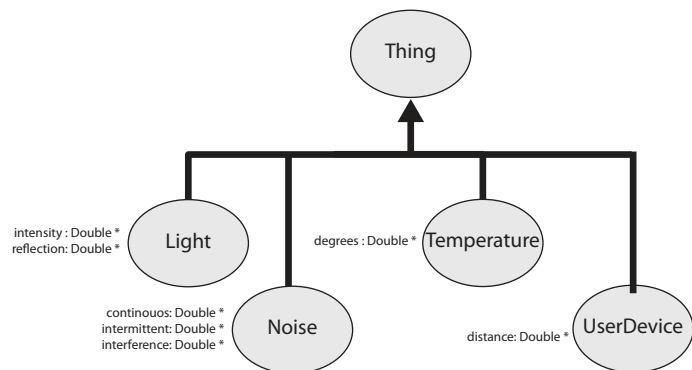


Figura 5.15: Ontología del ambiente físico.

Capítulo 6

Razonador Semántico en el Proceso de Visualización

Como se describió anteriormente, un razonador semántico tiene la capacidad de derivar consecuencias lógicas a partir de un conjunto de hechos o axiomas. Las reglas de inferencia utilizadas por el razonador se pueden expresar a través de un lenguaje de ontologías. Desde el punto de vista del proceso de visualización, un razonador semántico se podría utilizar para responder a las decisiones involucradas en el proceso de creación de una visualización, e.g. cuál es el mejor mapeo de color para cierto conjunto de datos.

A continuación describiremos las diferentes instancias en las que se utilizará al razonador dentro del proceso de visualización. Cabe aclarar que en todo momento en el que el razonador deriva información, debe contar con las reglas de inferencia apropiadas para dicha derivación.

6.1. Los datos

Los datos que ingresan al sistema no son sólo conjuntos de valores sino también relaciones entre ellos. Un razonador puede tomar la ontología que representa a los datos y derivar nuevas relaciones entre ellos (Figura 6.1), relaciones que el usuario quizás no sabía que existían. El razonador también puede completar la información existente con nuevos datos y finalmente es posible para el razonador chequear la consistencia de los datos de entrada.



Figura 6.1: El razonador semántico se puede utilizar para incorporar más información a los datos, ya sea derivando nuevas relaciones o completando los datos. Esta misma estrategia se puede utilizar para la semántica de los datos que ingresa al proceso.

6.1.1. Nuevas relaciones

Supongamos que en la semántica de los datos tenemos el concepto *Dato* y tres subconceptos del mismo: *Temperatura*, *Presión*, *Altura*. Estos tres elementos son disjuntos entre sí y representan el conjunto completo de *Dato*. Luego agregamos un nuevo subconcepto de *Dato* denominado *X* y establecemos que *X* es disjunto con *Presión* y *Altura*. La Figura 6.2 muestra el diagrama de conceptos involucrado. A partir de estos conceptos el razonador semántico puede derivar que *X* es en realidad un subconcepto de *Temperatura*, es decir surge una nueva relación entre *Temperatura* y *X*. Esto se aprecia en el diagrama 6.3 y se encuentra disponible en la web¹.

6.1.2. Completar los datos

En el contexto de la configuración de una técnica, podemos tener el concepto *Configuración* que contiene un atributo *clave* de tipo *string*. El valor de *clave* dependerá, en este caso, del tamaño del monitor a utilizar; por consiguiente, también incluiremos en este ejemplo el concepto *Monitor* con un atributo *tamaño* de tipo entero y con tres subconceptos: *Monitor de 15 pulgadas*, *Monitor de 17 pulgadas* y *Monitor de 19 pulgadas*. Cada uno de estos tres subconceptos posee una condición que asocia el tamaño del monitor (15, 17 y 19) al atributo *tamaño*. Como toda configuración tiene asociada un monitor, el concepto *Configuración* tiene una *object property* denominada *basadaEn* con dominio en *Configuración* y rango *Monitor*, siendo esta relación funcional.

¹<http://cs.uns.edu.ar/~mll/resources/thesis/nuevasrelaciones.owl>

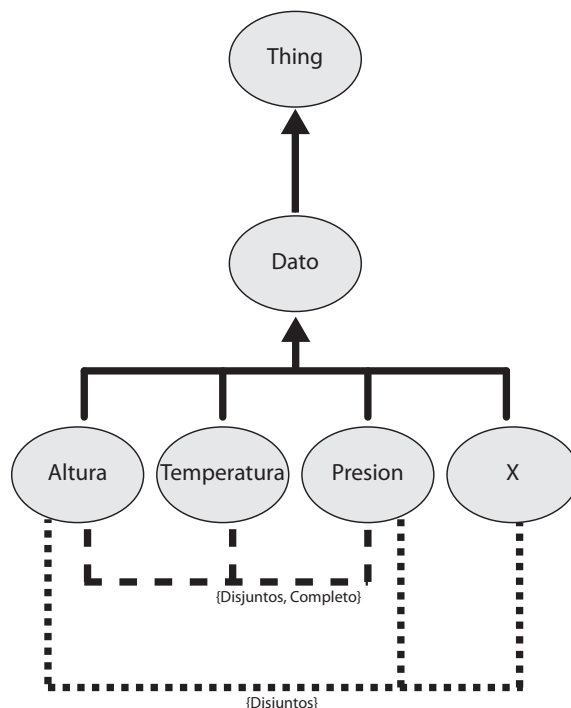


Figura 6.2: Ejemplo sobre el cual se ejecutará el razonador.

Para poder representar las diferentes configuraciones en función del tamaño del monitor, crearemos tres instancias de *Configuración* y tres de los monitores. Las tres instancias de los monitores serán: *m1* instancia de *Monitor de 15 pulgadas*, *m2* instancia de *Monitor de 17 pulgadas* y *m3* instancia de *Monitor de 19 pulgadas*. Al concepto *Monitor* se le agrega una condición que lo establece como un enumerado formado por las instancias *m1*, *m2* y *m3*. Las instancias de *Configuración* serán: *c1* cuyo atributo *clave* tendrá valor “bajo” y estará asociado a *m1*; *c2* cuyo atributo *clave* tendrá valor “medio” y estará asociado a *m2* y *c3* cuyo atributo *clave* tendrá valor “alto” y estará asociado a *m3*. En síntesis, tenemos tres tipos de configuraciones que dependen del monitor asociado. La Figura 6.4 representa los conceptos e instancias definidos.

A continuación realizaremos tres pasos: Primero crearemos una nueva instancia del concepto *Configuración* y la llamaremos *cx*; no le asignaremos ningún valor al atributo *clave*. Luego crearemos una instancia del concepto *Monitor de 15 pulgadas* llamada *mx* y finalmente asociaremos, a través de la *object property basadaEn* a *cx* con *mx*. En este punto podemos solicitarle al razonador que resuelva los valores de las propiedades de *cx*, lo que nos dará como resultado *valor = “bajo”*; esto se corresponde con la información que

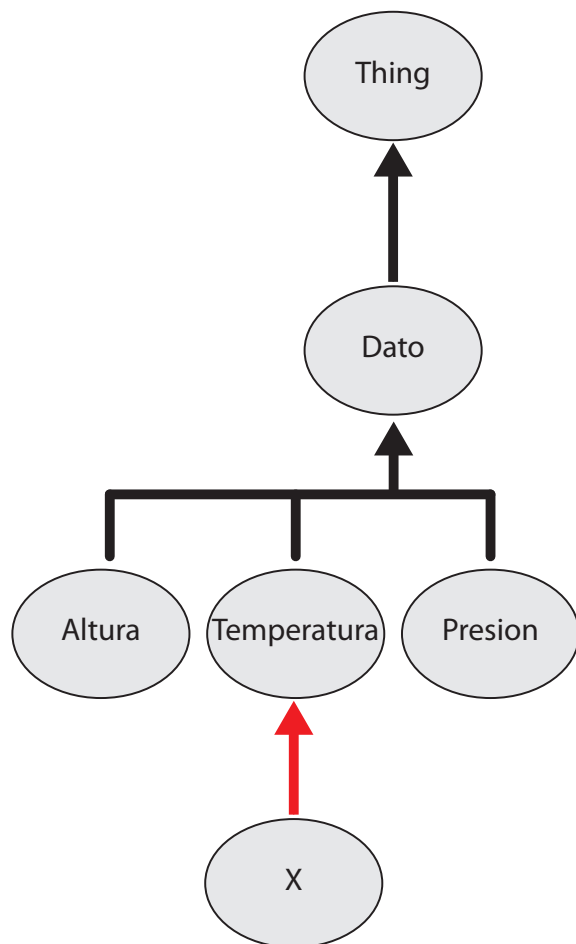


Figura 6.3: La relación descubierta por el razonador se encuentra marcada en rojo.

habíamos descrito anteriormente. De esta forma, el razonador completó la información de *cx* usando los conceptos e instancias establecidos. La Figura 6.5 muestra la información incorporada y el resultado del razonador. Este ejemplo se encuentra disponible en la web².

6.1.3. Chequeo de consistencia

Supongamos que, como parte de la semántica de los datos, se cuenta con los conceptos *Temperatura*, *TemperaturaAlta* y *TemperaturaBaja*; donde *TemperaturaAlta* y *TemperaturaBaja* son subconceptos de *Temperatura*. Como parte de la descripción de los datos se indica que *TemperaturaAlta* y *TemperaturaBaja* representan el conjunto

²<http://cs.uns.edu.ar/~mll/resources/thesis/completardatos.owl>

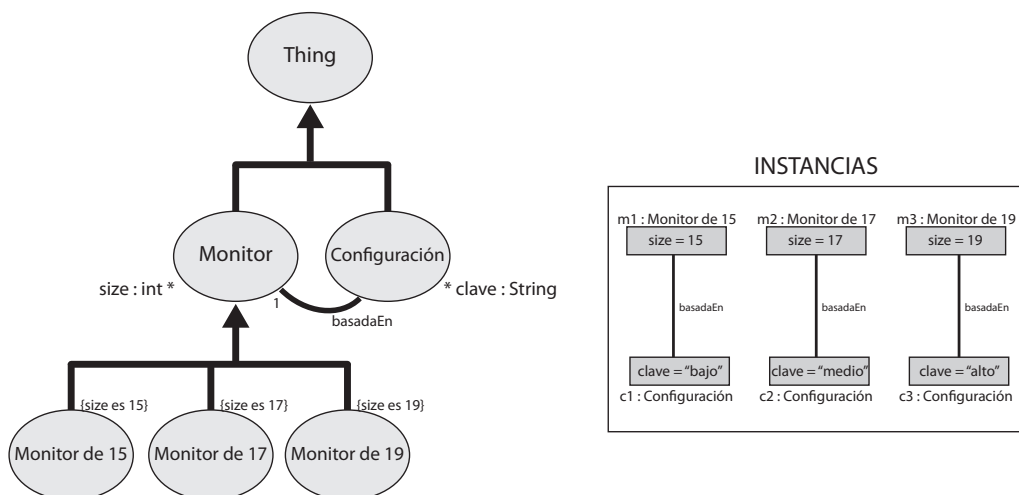


Figura 6.4: Ejemplo sobre el cual ejecutará el razonador.

completo de temperaturas y son disjuntos entre sí. Además, en esta semántica contamos con un concepto X , que también es subconcepto de *Temperatura* pero es disjunto con *TemperaturaAlta* y *TemperaturaBaja*. Esta última restricción genera una inconsistencia en la ontología debido a que se estableció que *TemperaturaAlta* y *TemperaturaBaja* son el conjunto completo de temperaturas. Debido a que X es disjunto con estas clases no puede ser una temperatura, sin embargo en la definición de X se indicó que es un subconcepto de *Temperatura*; dicha inconsistencia puede ser detectada por el razonador. La Figura 6.6 muestra gráficamente las relaciones entre los conceptos. Dicho ejemplo se encuentra disponible en internet³.

6.2. Las tareas

Al iniciar el proceso de visualización, el usuario establecerá cuáles son las tareas que quiere realizar sobre el resultado final. Estas tareas tendrán una semántica asociada y en caso de seleccionar múltiples tareas el razonador semántico podrá derivar posibles relaciones entre los elementos involucrados en las tareas (Figura 6.7). Por ejemplo, si el

³<http://cs.uns.edu.ar/~mll/resources/thesis/consistencia.owl>

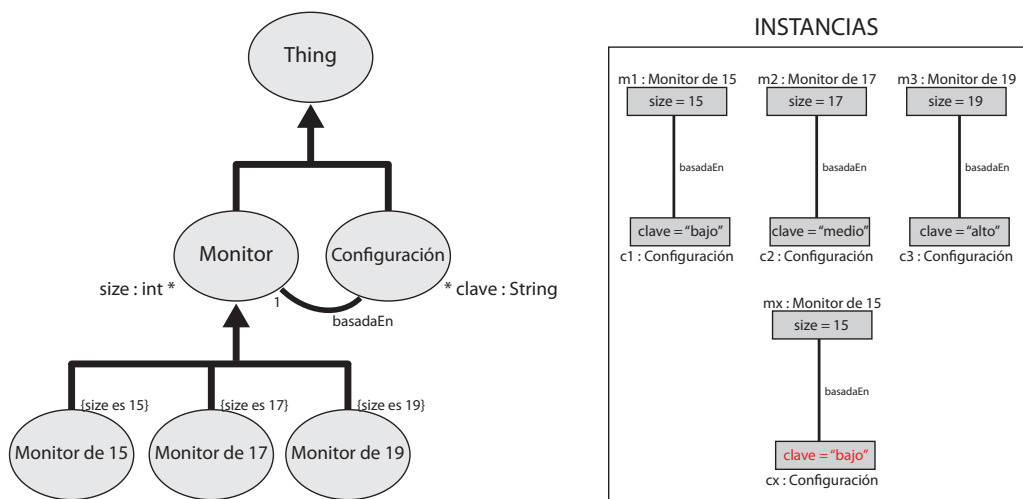


Figura 6.5: Ejemplo de cómo el razonador puede completar datos de una instancia. El texto en rojo corresponde a lo resuelto por el razonador.

usuario selecciona las tareas *agrupamiento* y *ranking*, el razonador derivará la relación *cada grupo tendrá un ranking asociado*.

6.3. El Hardware

Tanto las tareas como los datos se pueden considerar como datos de entrada dados por el usuario al proceso de visualización (en las Figuras 6.1 y 6.7 tanto las tareas como los datos “entran” al proceso de visualización). Sin embargo también podemos considerar aquellos elementos que no serán descritos por el usuario sino capturados por el sistema. En este caso consideramos el *hardware* sobre el que se ejecuta el proceso de visualización. El proceso puede tomar las características del *hardware* y obtener, a través del razonador, una descripción de *hardware* y sus capacidades. Por ejemplo, el razonador podría derivar qué tanta capacidad tiene el sistema para procesar gráficos en 3D en función del tipo de GPU.

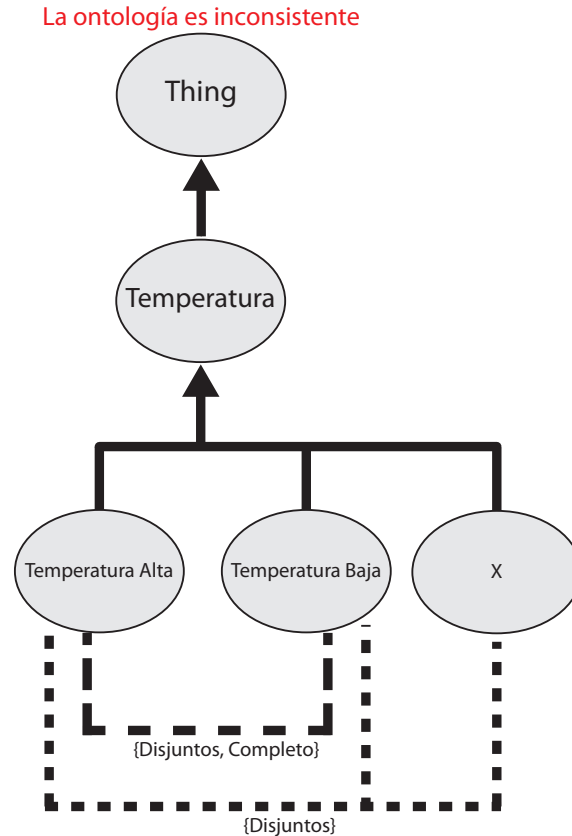


Figura 6.6: Ejemplo de una ontología inconsistente.

6.4. La técnica de representación

Como se dijo antes, la selección de la técnica de representación es vital. Una mala elección puede producir una visualización que no permita al usuario cumplir las tareas deseadas, o hacerlo en una forma no efectiva. La selección de la técnica más adecuada se puede realizar a través del razonador semántico. Sin duda, los elementos involucrados en este proceso son: la ontología de datos extendida, el resultado del razonador sobre las tareas, la ontología de las técnicas y la semántica del usuario. Sin embargo, el hardware también es un elemento clave. Si se aplica una técnica 3D sobre un hardware que no es apropiado para esto, por ejemplo un dispositivo que no posea una GPU adecuada, los tiempos de *rendering* e interacción no serían los apropiados. Es por esto último que la ontología del hardware también estará involucrada en este proceso (Figura 6.9). De la semántica del usuario será requerida la información asociada a habilidades y capacidades;

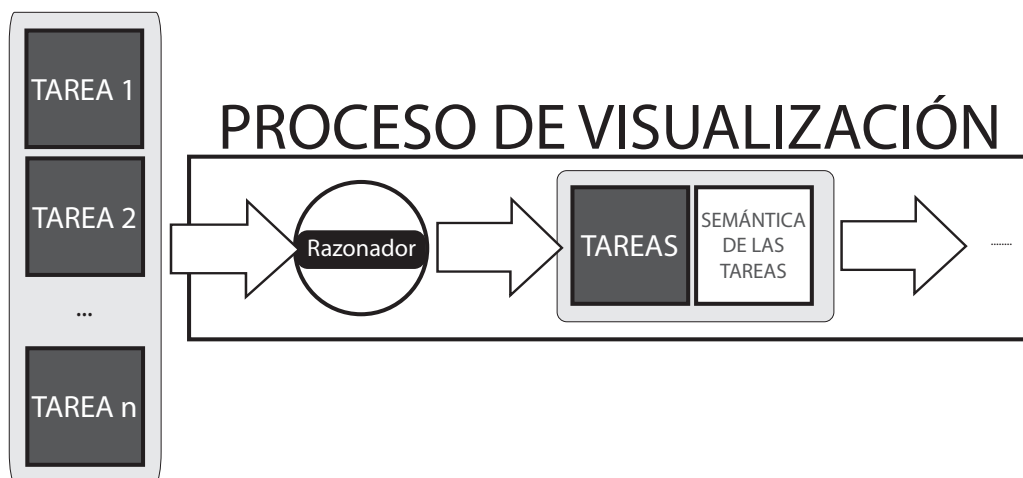


Figura 6.7: El razonador semántico se puede utilizar para relacionar información entre tareas. Si el usuario desea realizar más de una tarea, el razonador semántico puede descubrir relaciones entre ellas.

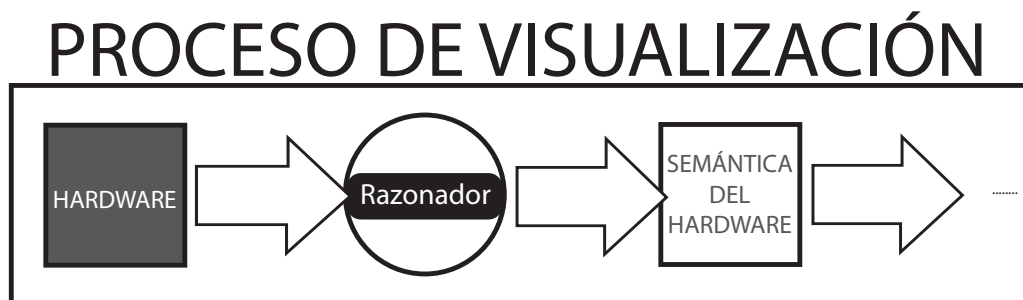


Figura 6.8: El razonador semántico puede derivar una descripción de las capacidades del sistema conociendo sus características.

por ejemplo, en tanto sea posible no se debería elegir una técnica 3D si el usuario no puede percibir profundidad. El resultado de este proceso es la técnica que mejor se ajusta para visualizar el conjunto de datos de entrada, para poder realizar las tareas especificadas bajo el hardware actual.

6.5. El mapeo visual

El mapeo visual se define en función de la técnica a utilizar y las características de los datos. Por ejemplo, si la técnica requiere un mapa de color es necesario saber si los datos

PROCESO DE VISUALIZACIÓN

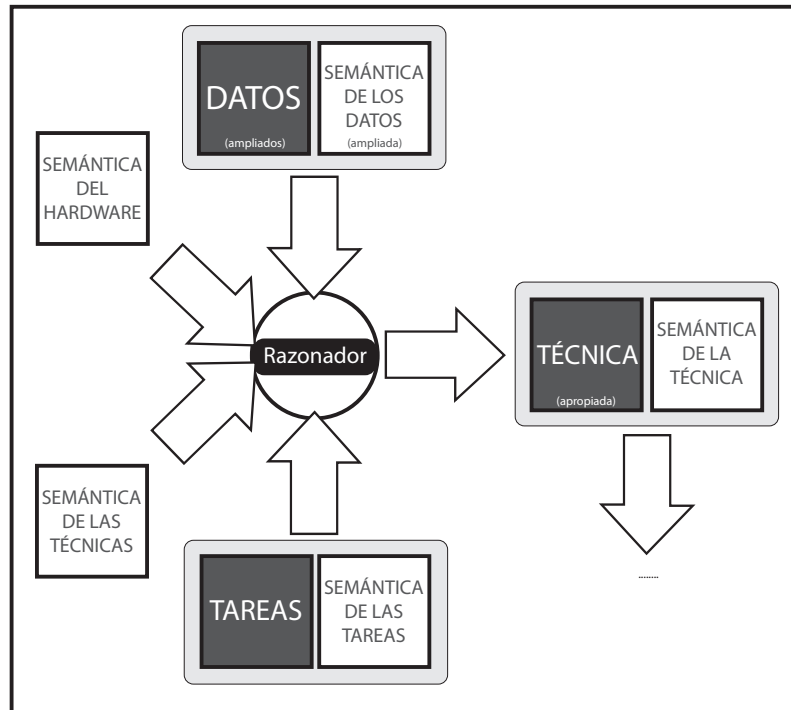


Figura 6.9: El razonador semántico puede determinar cuál es la técnica más apropiada para representar los datos ingresados, para las tareas especificadas sobre el *hardware* presente.

que serán representados con color son de tipo ordinal o razón. La semántica de los datos también servirá para definir el color a utilizar; si se está representado una temperatura es lógico pensar en un mapa de color que incluya los colores rojo, para temperaturas altas y azul para temperaturas bajas. Sin embargo, esta última asignación, “temperaturas altas” con color “rojo” es un ejemplo de cómo el modelo cultural del usuario define las características visuales. El razonador semántico también debe considerar la semántica del contexto y la semántica del humano al momento de inferir cuál es el mapeo visual apropiado (Figura 6.10). De la semántica del contexto se requiere la semántica asociada al dominio de aplicación y al dominio organizacional; de la semántica del humano se requiere su modelo cultural, sus habilidades y capacidades. Tanto la semántica del dominio de aplicación, dominio organizacional y modelo cultural definen mapeos de conceptos a características visuales y es posible que existan colisiones de conceptos. Es decir, un mismo concepto puede tener diferentes representaciones de acuerdo al modelo cultural, dominio

de aplicación o dominio organizacional; por el momento nuestra propuesta para manejar estas colisiones es solicitándole al usuario que elija qué mapeo quiere priorizar. En esta etapa también es necesario conocer las habilidades y capacidades del usuario; no se puede crear un mapeo visual en base a colores que el usuario no puede percibir correctamente.

PROCESO DE VISUALIZACIÓN

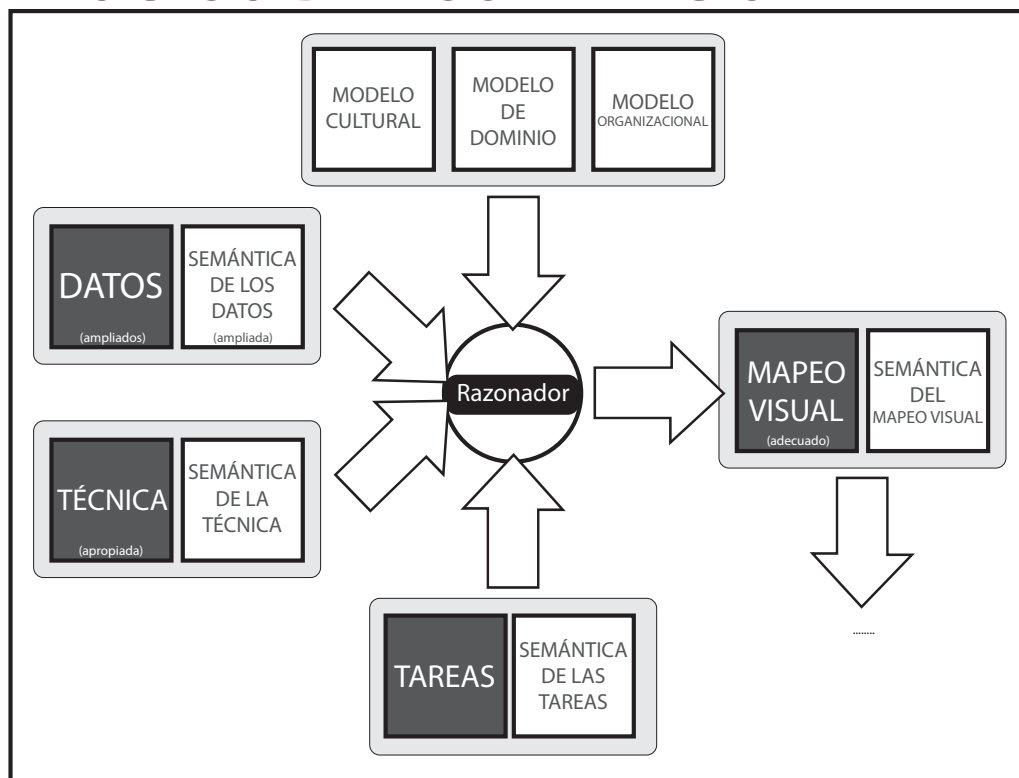


Figura 6.10: El razonador semántico puede determinar cómo generar el mapeo visual más apropiado. El mapeo visual depende de la técnica seleccionada, de las características de los datos y del modelo cultural asociado al usuario.

6.6. La visualización

Una vez establecido cómo van a ser representados los datos, se debe crear la visualización correspondiente. En este paso también se puede involucrar al razonador. Resolver qué característica o qué elemento visual utilizar para cada dato puede ser responsabilidad del razonador. Para poder lograr la visualización el razonador requerirá de

los datos y su semántica, la técnica seleccionada y su semántica y finalmente el mapeo visual seleccionado y su semántica. La Figura 6.11 muestra los elementos involucrados en esta etapa del proceso.

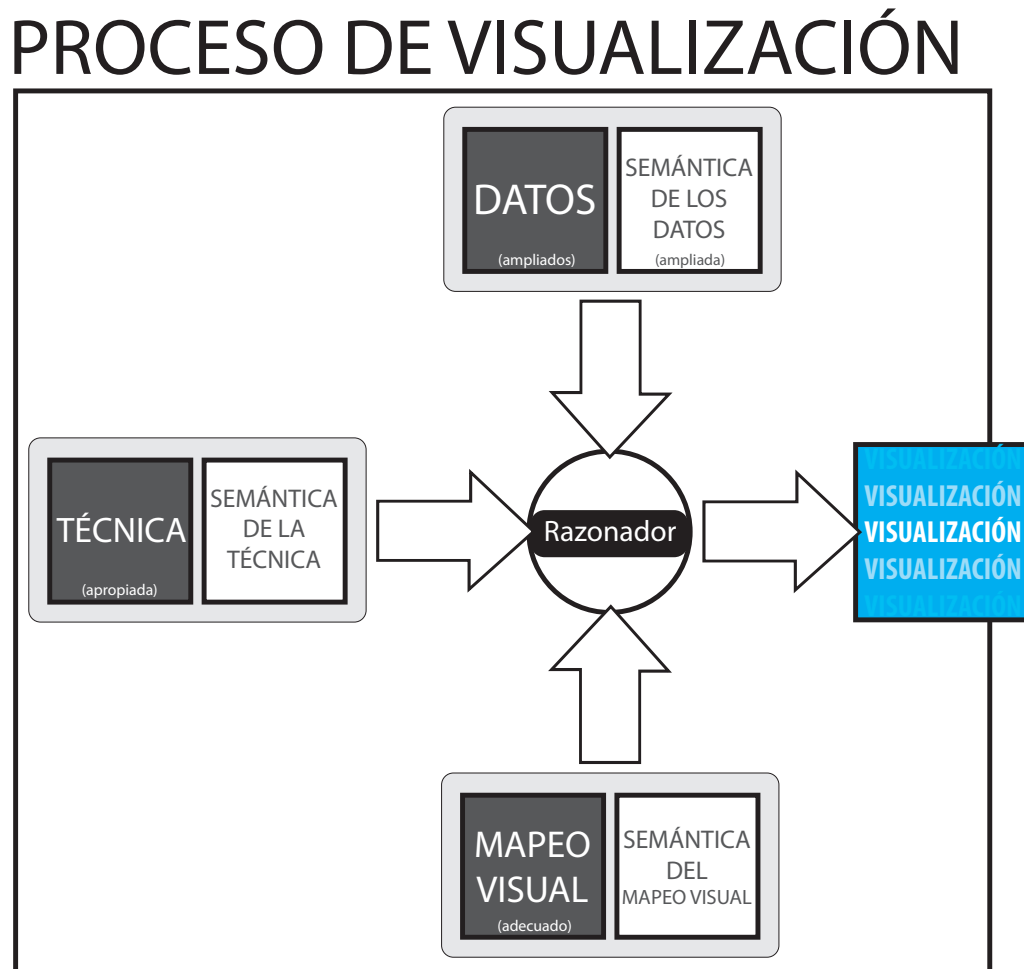


Figura 6.11: El razonador semántico puede ser utilizado para la creación *per se* de la visualización.

6.7. El proceso de visualización

La Figura 6.12 muestra la composición integrada de las etapas descritas previamente.

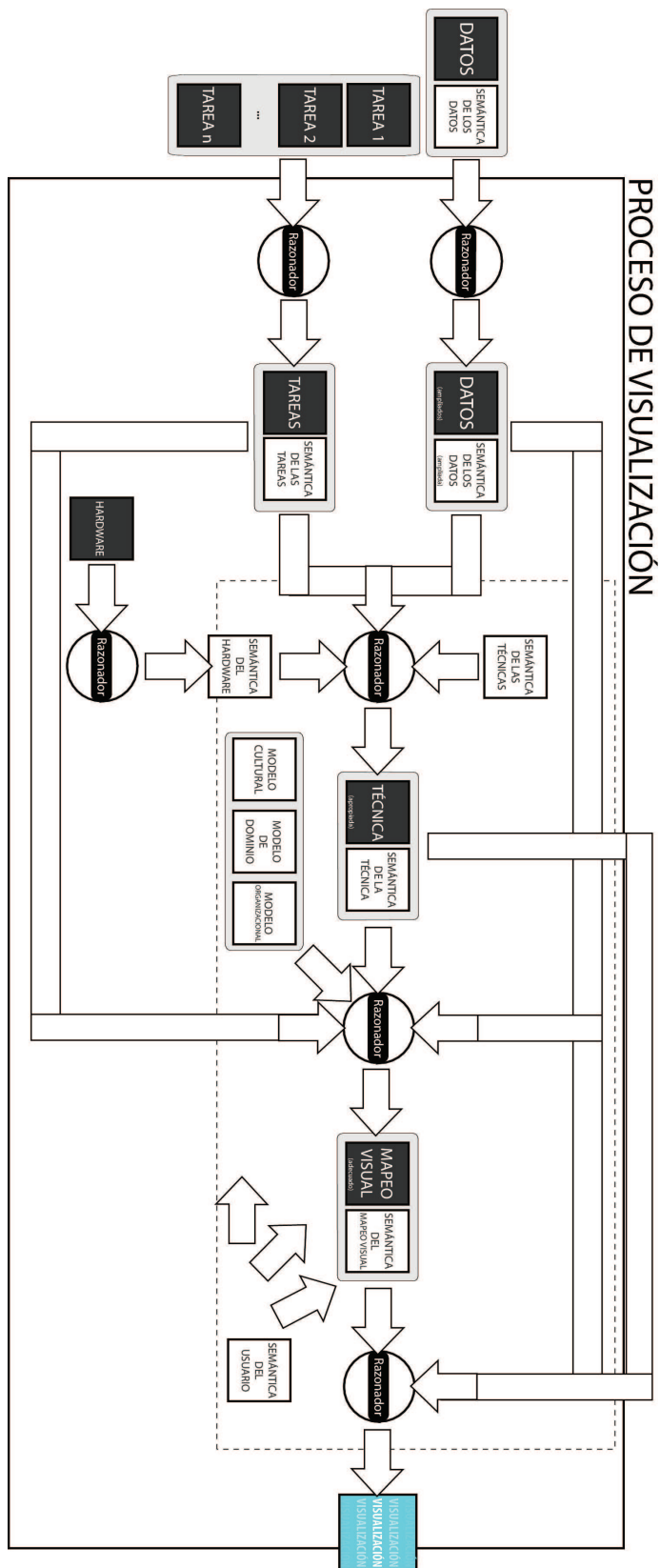


Figura 6.12: Uso del razonador semántico a través del proceso de visualización.

Capítulo 7

Casos de Estudio

Con el objetivo de validar la arquitectura presentada se desarrolló una implementación parcial de la misma. En este capítulo se detallan los prototipos con los cuales se implementó una visualización basada en semántica para una aplicación de visualización de estructuras de directorios. En esta implementación se han incluido los elementos necesarios para mostrar cómo es posible utilizar información semántica como ayuda y guía en el proceso de creación de una visualización.

7.1. Brows.AR: Visualizador de jerarquías de archivos

El diagramado Esférico ([Lar06], [LMC07b], [LUMC09]) es una generalización en 3D del diagramado Radial. En el diagramado radial los nodos se ubican sobre el perímetro de una circunferencia y en el esférico estos son ubicados sobre la superficie de esferas concéntricas. Para validar dicho diagramado desarrollamos una aplicación para la visualización de jerarquías de archivos, llamada Brows.AR (Figura 7.1).

Brows.AR es una aplicación multiplataforma desarrollada en Java que permite visualizar, utilizando el diagramado Esférico, la jerarquía de archivos y carpetas asociada a una carpeta o unidad particular. Además de utilizar el diagramado Esférico, esta aplicación permite variar la forma en cómo se representa visualmente la jerarquía.

- **Nodos.** Los nodos pueden ser representados por un punto en el espacio, un cubo o una esfera. La única propiedad visual de los puntos es el color; éstos son los elementos

menos complejos de la técnica. La esfera es el elemento visual más complejo, seguido por el cubo. También es posible no tener un elemento visual asignado a los nodos, es decir no mostrarlos. Esto nos da cuatro posibilidades de representación de los nodos: no asignadas, puntos, cubos o esferas.

- **Arcos.** Los arcos pueden ser representados por dos tipos de líneas: una sola línea cuya única propiedad visual es la de color o un cilindro, que permite incorporar propiedades visuales adicionales. Este último es el elemento visual más complejo. También es posible no tener un elemento visual asignado a los arcos. Por lo tanto, para los arcos, tenemos tres posibles representaciones: no asignada, líneas o cilindros.
- **Ayuda Visual.** En esta implementación del diagramado Esférico los nodos están distribuidos uniformemente en la superficie de esferas discretizadas con triángulos. Los nodos son ubicados en el baricentro de algunos de estos triángulos. Como una ayuda visual, en la visualización es posible mostrar los triángulos que definen la ubicación de los nodos.

Con Brows.AR también es posible, para el caso de jerarquías con un factor de ramificación o cantidad de nodos superior a una constante definida, no mostrar los nodos ni los arcos y sólo dejar los triángulos. De esta forma se puede tener un *overview* de la jerarquía sin comprometer la *performance* de la aplicación. Para mayores detalles sobre el diagramado Esférico y la implementación de Brows.AR, el lector puede ver [LMC07b].

7.2. Primer Prototipo. Asignación de Color Basada en Semántica

El primer prototipo ([LMC10a]) que fue desarrollado se centró exclusivamente en la asignación del atributo color a los elementos visuales. Al asignarle color a elementos visuales es importante que elementos diferentes se identifiquen con colores diferentes, si un desarrollador no respeta esto, el resultado sería una visualización que transmitiría información incorrecta. Al resolver este problema mediante semántica liberamos al desarrollador de esta decisión y aseguramos que la asignación de color sea siempre la apropiada. En este prototipo, la asignación de color se resolvió utilizando ontologías y un razonador semántico.

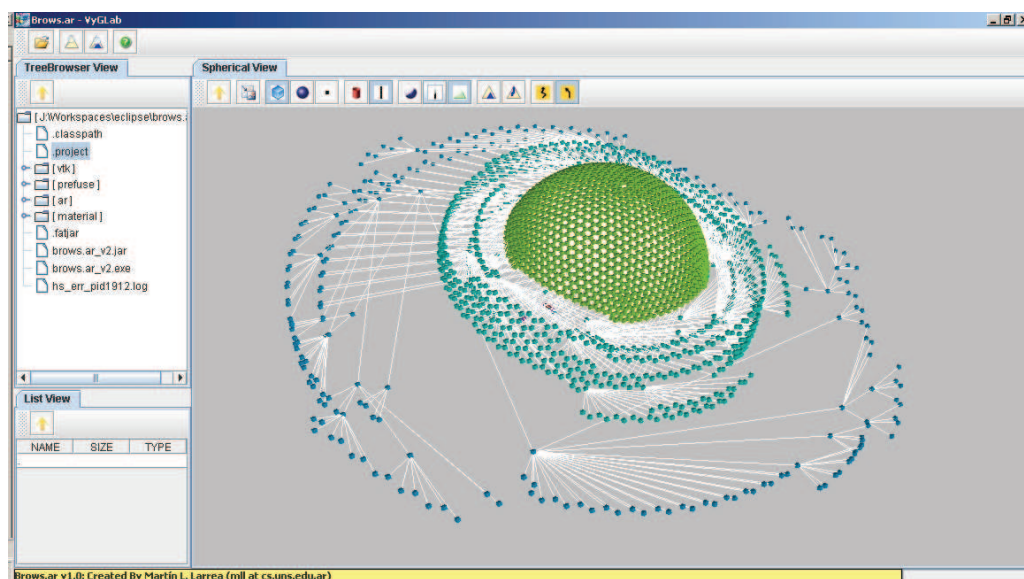


Figura 7.1: Captura de pantalla de Brows.AR, una herramienta para la visualización de jerarquías de archivos utilizando el diagramado Esférico ([Lar06], [LMC07b], [LUMC09]).

Este trabajo se realizó utilizando Brows.AR como aplicación base. Al utilizar Brows.AR, los datos a representar son archivos y carpetas dentro de una unidad de almacenamiento. Con este trabajo se buscó describir, a través de ontologías, la relación existente entre los datos a visualizar y el color utilizado para su representación. En particular, para este tipo de aplicación, se representó la asociación entre los tipos de archivos y carpetas con los colores usados para representarlos visualmente.

7.2.1. Protégé, Jena & Pellet

Para poder trabajar con ontologías a nivel de código Java se decidió utilizar las librerías de Protégé-OWL ([GMF⁺03], [KMR04], [KFNM04]) y Jena ([CDD⁺04], [Gro09] [McB02]). Protégé-OWL es una extensión del editor Protégé, la cual brinda soporte para OWL. La API de Protégé-OWL ofrece la misma funcionalidad que el editor pero a través de librerías Java. La API de Protégé-OWL está estrechamente ligada a Jena. Jena es un framework Java para construir aplicaciones para la Web Semántica. Proporciona un entorno de programación para RDF, RDFS, OWL y SPARQL. La decisión de utilizar estas librerías se basó en que tanto la API Protégé-OWL como la de Jena son *open-source*, poseen una

amplia comunidad de soporte en la Web y son reconocidas herramientas en el ambiente académico y comercial.

Para el razonador se utilizó la API de Pellet ([SPG⁺07b]). Pellet es un razonador gratuito disponible en la Web, soporta OWL-DL con muy buena *performance*, es el primer razonador OWL-DL sensato y completo y es uno de los más completos, hasta el momento, en cuanto a las características soportados.

7.2.2. Implementación

Se comenzó creando la ontología *VisualColorMapping*, que contiene los conceptos necesarios para el proceso de visualización. Se introdujo el concepto *Color*, que representa el color que se utilizará para pintar los elementos visuales. El sistema de color que se decidió utilizar es el de RGB, por lo tanto se incluyeron tres *datatype properties* de tipo entero con valores entre 0 y 255; estas propiedades se denominaron *red*, *green* y *blue*, y tienen como dominio el concepto *Color*.

Para representar los datos a visualizar se creó el concepto *Data*. Se crearon dos *datatype properties*, *description* que representa una breve descripción del dato y *filetype* que indica la extensión del archivo asociado al dato. Ambas propiedades son de tipo *string* y tienen como dominio el concepto *Data*. El vínculo entre *Data* y *Color* se realiza a través de un tercer concepto denominado *VisualRepresentation*. El vínculo entre los tres conceptos puede leerse como “*every data element has a visual representation which has a color property*”. Sin duda el color no es la única propiedad visual que se le puede asociar a una representación visual; sin embargo, en este prototipo nos limitaremos solamente al color.

Para poder vincular los tres conceptos presentados utilizaremos *object properties*. El primer *object property functional* será *inRepresentation*, el cual tiene como dominio a *Data* y rango a *VisualRepresentation*. El segundo *object property functional* será *inColor* el cual tiene como dominio a *VisualRepresentation* y rango *Color*. La Figura 7.2 muestra gráficamente la ontología.

La información que contiene la ontología actual no es suficiente para que el razonador opere. Para lograr obtener resultados a partir del razonador es necesario incorporar más información a la ontología. Para cada tipo de archivo que se desee visualizar crearemos un nuevo concepto que tendrá como superclase a *Data*. Estos nuevos conceptos tendrán

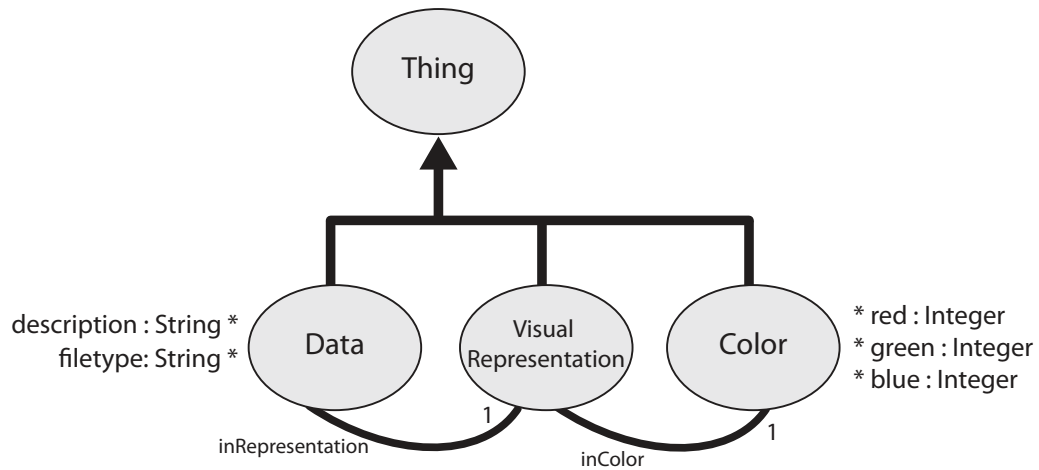


Figura 7.2: Diagrama de la ontología *VisualColorMapping* para el primer prototipo ([LMC10a]).

una condición necesaria y suficiente del tipo *filetype has extension*, donde *extension* es la extensión que queremos representar y *filetype* es la propiedad heredada del concepto *Data*. Supongamos que queremos representar archivos “pdf” con el color rojo. Para representar eso crearíamos un nuevo concepto llamado *PDF* con superclase *Data*. En este nuevo concepto incluiríamos la condición necesaria y suficiente en la forma de *filetype has “pdf”* y la propiedad *description* con el valor “*PDF Document*”. Para poder representar este elemento con el color rojo crearemos el concepto *RED* con superclase *Color* y valores 255, 0 y 0 para sus propiedades *red*, *green* y *blue* respectivamente. El siguiente paso puede resultar no tan intuitivo como los anteriores. Hasta el momento se crearon dos nuevos conceptos, uno con superclase *Data* y el otro con superclase en *Color* y se vincularon a través de *VisualRepresentation*. En este paso es necesario crear un nuevo concepto llamado *VisualRed* con superclase *VisualRepresentation*. Este nuevo concepto servirá como vínculo entre el concepto *PDF* y *RED*. También es necesario crear las subpropiedades de *inRepresentation* y *inColor*, denominadas *inRepresentationRed* y *inColorRed* respectivamente. El dominio de la propiedad *inRepresentationRed* será *PDF* y su rango *VisualRed*. El dominio de *inColorRed* será *VisualRed* y su rango *RED*. La Figura 7.3 muestra lo descrito hasta el momento.

Para cada nuevo tipo de archivo *ftype* que deseemos representar utilizando el color *c* se deben crear tres nuevos conceptos; uno para *ftype*, otro para *c* y finalmente uno, con superclase *VisualRepresentation*, que los vincule. También será necesario crear dos nuevas

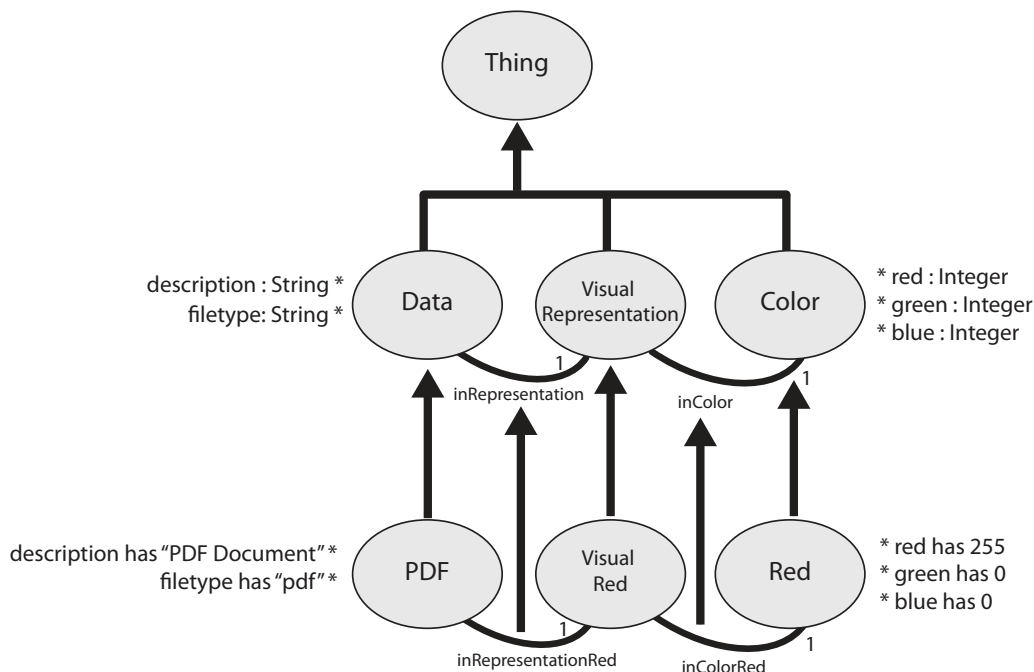


Figura 7.3: Diagrama de la ontología *VisualColorMapping* para el primer prototipo. Inclusión de información sobre cómo los archivos “pdf” son representados con el color rojo ([LMC10a]).

subpropiedades, una de *inRepresentation* y otra de *inColor*. Definido el contenido de la ontología, podemos enfocarnos en cómo la utilizará el razonador semántico.

A continuación describiremos los pasos a seguir para poder determinar el color que se utilizará para representar un archivo. Llamaremos *fctype* a la extensión asociada al archivo que representaremos visualmente. Comenzaremos creando un subconcepto de *Data* al que llamaremos *temp* y tendrá una condición necesaria y suficiente de la forma *filetype has fctype*. En este punto se le solicita al razonador que obtenga los conceptos derivados. Si en nuestra ontología *VisualColorMapping* existe un concepto definido para *fctype* entonces el razonador retornará dicho concepto, caso contrario retornará *owl:Nothing*. Si el concepto retornado, al que llamaremos *Cftype*, es diferente de *owl:Nothing* entonces podemos obtener el color asociado siguiendo las *object properties inRepresentation* y *inColor*.

Para poder integrar este proceso a la aplicación Brows.AR se creó la clase *Reasoner* la cual posee un método, *ask*. Este método toma como parámetro de entrada la extensión de un archivo y retorna un color. Este color será el usado para pintar el elemento

visual que represente dicho archivo. La clase *Reasoner* utiliza las API de Protégé y Jena para interactuar con la ontología. El método *ask* implementa en código Java los pasos descritos anteriormente. El constructor de la clase *Reasoner* toma un parámetro de entrada, un *JenaOwlModel* el cual es una representación de una ontología. Para mejorar la performance se utilizó una tabla Hash a modo de memoria caché para almacenar los resultados obtenidos del razonador. Cuando se ejecuta el método *ask* con un tipo de archivo *ftype*, primero se chequea en la tabla Hash si esa consulta ya fue resuelta; en caso que no haya sido resuelta, se utiliza el razonador para obtener el color asociado *c*. Antes de retornar el color *c*, el método *ask* almacena el par *(ftype, c)* en la tabla Hash. Si al ejecutar el método *ask* el parámetro *ftype* sí se encuentra en la tabla Hash, entonces se retorna el color asociado.

```

1. OWLNamedClass DataClass =
    owlModel.getOWLNamedClass("Data");
2. OWLNamedClass tempSubClass =
    owlModel.createOWLNamedSubclass("temp", DataClass);
3. OWLIntersectionClass t =
    owlModel.createOWLIntersectionClass();
4. OWLDatatypeProperty fileTypeProperty =
    owlModel.getOWLDatatypeProperty("filetype");
5. OWLHasValue hs =
    owlModel.createOWLHasValue(fileTypeProperty, filetype);
6. t.addOperand(hs);
7. tempSubClass.setDefinition(t);
8. Collection inferredSubclasses =
    this.pelletReasoner.getSubclasses(tempSubClass);
9. Iterator it = inferredSubclasses.iterator();
10. OWLNamedClass inferredSubclass =
    (OWLNamedClass) it.next();

```

Figura 7.4: Código principal del método *ask* en la clase *Reasoner*. Las líneas 1 a 3 crean un concepto con superclase *Data*. Las líneas 4 a 7 agregan la condición necesaria y suficiente “*filetype has ftype*”. La línea 8 consulta al razonador por las subclases inferidas. Las líneas 9 y 10 obtienen dichas subclases ([LMC10a]).

Sin duda este primer prototipo es muy limitado. El principal problema que se observa es que la información de los datos a visualizar y la forma de visualizarlos se encuentran en una sola ontología. Sería preferible mantener estas dos áreas separadas para mejorar la flexibilidad de la solución. También es necesario incorporar nuevos elementos al mapeo visual, por ejemplo formas.

7.3. Segundo Prototipo. Creación de una Visualización utilizando Información Semántica

En el segundo prototipo, se mejoró la arquitectura del sistema y se incorporaron nuevos elementos. En esta nueva versión se resuelve el problema de la asignación de color pero ahora se realiza a partir de la información semántica de la tarea; además se incorpora la configuración de la técnica de visualización, también a partir de información semántica. Se agregaron nuevas ontologías y se utiliza el razonador semántico en distintos puntos adicionales del proceso de visualización. Esta segunda implementación también se realizó sobre Brows.AR. La integración con la aplicación Brows.AR se realizó de igual manera que en el caso anterior. A diferencia del prototipo anterior, esta implementación cuenta con cuatro ontologías: *Datos*, *Tarea*, *Hardware* y *Color*. A continuación se describe la implementación.

7.3.1. Ontología de Color

En esta implementación, nuestra ontología de Color se corresponde con la ontología presentada en 5.1.5. También se incluyó la ontología de *Colores* la cual contiene, en esta versión, 16 colores.

7.3.2. Ontología de Datos

En esta ontología sólo se incluyó información basada en métricas sobre los datos. Teniendo un solo concepto, *Metrics*, se crearon 5 *data properties* con dominio en el concepto *Metrics*. Debido a que Brows.AR se utiliza para visualizar jerarquías, las métricas utilizadas son orientadas a árboles. Las métricas utilizadas fueron:

- Cantidad de elementos (n). En este caso, n representa la cantidad de archivos y carpetas a visualizar.
- Altura del árbol (h). Cantidad de elementos en el camino más largo desde la raíz hasta una hoja.
- Ancho del árbol (w). Cantidad máxima de elementos en un nivel del árbol.

- Razón del árbol. Corresponde a la medida $r = Height/Width$.
- *Bounding box* del árbol. Corresponde a la medida $bb = Alto * Ancho$.

7.3.3. Ontología de Tareas

En esta nueva implementación incluimos la semántica de las tareas. La aplicación Brows.AR sólo soporta dos tareas, exploración y filtrado. En el caso de exploración el usuario puede navegar el espacio 3D creado en la visualización; el filtrado se realiza sobre los tipos de archivos, es decir, el usuario puede pedir aplicar un filtrado sobre los archivos de extensión “pdf”.

Tal como se describió en 4.2.2 y 5.3 la exploración no implica una modificación de la forma en cómo se representan los datos por lo que, a nivel de la ontología, no tendrá representación. Sin embargo, no es éste el caso para filtrado. Filtrado se define como: *given some conditions on attributes values, select data cases satisfying those conditions* ([LPP⁺06]). En nuestro prototipo resaltaremos los elementos seleccionados cambiando su color; esto hará que resalten con respecto a los demás elementos. Se planteó cómo desafío describir, en una ontología, la relación entre los colores y los elementos filtrados.

Para resaltar un elemento utilizando color es necesario conocer cuál es el color del fondo, e igual de importante es saber qué color usar sobre aquellos elementos que no serán resaltados. Para representar estos elementos incluimos tres conceptos en la ontología: *Background*, *Highlight* y *Regular*. El concepto *Background* es dominio de dos *object properties* que lo vinculan con los conceptos *Highlight* y *Regular*, tales propiedades son *toHighlight* y *toRegular*.

Aún no hemos establecido que el filtrado se realizará utilizando colores. Para hacerlo se indica que tanto *Background*, *Highlight* y *Regular* son equivalentes al concepto *Color*. Para asegurarnos que el color utilizado para resaltar efectivamente resalte sobre el color de fondo se indicó que la propiedad *toHighlight* es equivalente a la *object property opposite* y *toRegular* es equivalente a *next*, ambas de la ontología de color. De esta manera estamos indicando que el filtrado se mostrará a través de color y que el color utilizado para resaltar es el color opuesto al utilizado de fondo. El mayor beneficio de esta representación es que el usuario ya no es responsable de la selección de colores y que ésta la realiza el mismo

proceso de visualización a través del razonador y la información semántica; una mala selección de colores por parte del usuario podría resultar en una visualización donde los elementos resaltados no se vean como tales debido a que el contraste entre el color de fondo y el color utilizado para resaltar no es perceptible. Otra ventaja importante es que es muy fácil cambiar la manera de mostrar el filtrado.

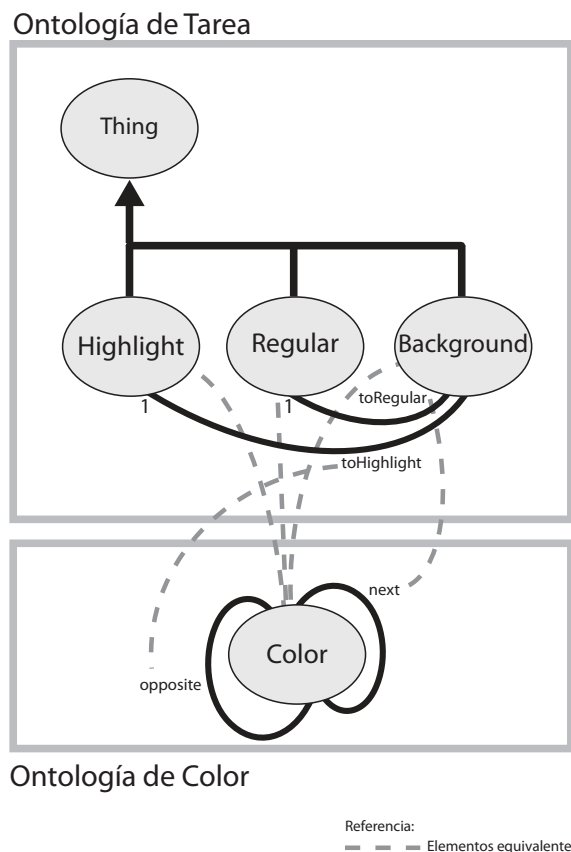


Figura 7.5: Ontología de *Tarea*. En este prototipo se describe solamente la tarea *Filtrado* y cómo se vincula con la ontología de *Color*

7.3.4. Ontología de Hardware

Como se mencionó previamente, una visualización no ocurre en forma aislada del mundo; una visualización ocurre en un contexto. En este caso consideramos que tal contexto es el hardware sobre el cual se está ejecutando la visualización. La ontología

de hardware contiene un concepto denominado *Hardware* y cinco *data properties* cuyo dominio es tal concepto. Las propiedades son:

- Un indicador de si se cuenta o no con una aceleradora gráfica (*GPU*). Esta propiedad es de tipo boolean.
- La altura, en pixels, de la resolución actual (*hp*). Esta propiedad es de tipo integer.
- El ancho, en pixels, de la resolución actual (*wp*). Esta propiedad es de tipo integer.
- La cantidad de pixels de la resolución actual, ($pxs = hp * wp$). Esta propiedad es de tipo integer.
- El tamaño, en pulgadas, del monitor (*inches*). Esta propiedad es de tipo real.

7.3.5. Proceso de razonamiento

A continuación se describirá cómo se utilizó el razonador semántico para, utilizando las ontologías antes descritas, generar resultados que permitirán automatizar el proceso de creación de la visualización.

Asignación de color de acuerdo a la semántica de la tarea de filtrado

El primer paso para resolver la asignación de colores a elementos visuales es el de determinar la paleta de colores que se utilizará. Para calcular estos valores, la aplicación le solicita al usuario que seleccione el color de fondo que desea utilizar para la visualización. Las opciones de colores que se le presentan al usuario se corresponde con los 16 colores representados en la ontología de colores. El usuario selecciona 1 de los 16 colores, el cual se almacena en la aplicación. El color seleccionado junto con su vecino y opuesto forman el esquema de colores; tanto el vecino como el opuesto se obtienen a partir de la ontología de colores y las propiedades *next* y *opposite*.

El siguiente paso es tomar el esquema de colores y crear el mapa de asignación de colores. El mapa de asignación de colores representa la forma en la cual se utilizarán los colores, es decir qué color se utilizará para qué. En este punto se utiliza el razonador para determinar esto. Como el concepto *Color* es equivalente al concepto *Background* de la ontología de Tarea, todo lo que sea verdad para *Color* y para *Background* lo será para el

color seleccionado de fondo, debido a que éste es una instancia de ambos conceptos. Al obtener las subclases derivadas, el razonador obtiene el equivalente de *opposite*, es decir el color que corresponde a través de la propiedad *highlight*. Recordemos que se especificó en la ontología de tareas que *opposite* y *highlight* son equivalentes. Gracias a esto, cualquiera sea la instancia del concepto *Color* que se encuentre como rango de la propiedad *highlight* será utilizada para pintar los elementos que se deban resaltar en la visualización. Estos mismos pasos se repiten sobre las propiedades *next* y *regular*.

Conocer qué color se utilizará para resaltar ciertos elementos y cuál para los demás elementos es lo que se denominó “mapa de asignación de colores”. Terminado este proceso, el siguiente consiste en determinar cómo se configurará la visualización en función de la semántica de los datos, el hardware y la tarea.

Configuración de la visualización utilizando semántica

Tal como se describió en el prototipo anterior, Brows.AR permite variantes sobre cómo representar gráficamente los elementos visuales. En este nuevo prototipo se mantienen las mismas variantes a nivel de representación de nodos, arcos y ayudas visuales. Para definir cómo configurar la visualización creamos una serie de reglas que vinculan la semántica de los datos, las tareas y el hardware con las opciones soportadas por la aplicación para la visualización. La Figura 7.6 muestra el conjunto de reglas definido. La condición *isTaskFilter* se refiere a si el usuario está realizando la tarea de filtrado o no. Recordemos que la tarea de exploración no afecta el mapeo visual. El resultado de este proceso será una instancia de un concepto denominado *Configuration* el cual indica qué características visuales utilizar en la visualización. Este conjunto de reglas nos permite controlar cómo la visualización es creada para lograr el mejor resultado posible. Como puede verse en la Figura 7.6, se requiere un preprocesamiento de la información semántica. Por ejemplo, en lugar de preguntar si la cantidad de datos es superior a un número dado, se consulta si la cantidad de datos es “large”. Esto se hace para evitar usar comparaciones de tipo mayor o menor y así poder simplificar el proceso de razonamiento. Actualmente se está completando el mapeo de estas reglas a una ontología que pueda ser utilizada por el razonador para derivar la configuración apropiada.

Generación de la Visualización

El último paso en el que se utiliza el razonador es en la creación de la visualización *per se*. Los elementos que son requeridos por el razonador son: la configuración de la visualización (una instancia del concepto *Configuration*), el mapa de asignación de colores, los datos y la semántica de la tarea si corresponde. Tal como se hizo en [LMC10a] el razonador puede determinar qué color usar para qué dato en base al mapa de asignación de colores. Utilizando la configuración de la visualización, el razonador puede determinar si corresponde utilizar las ayudas visuales o no. La arquitectura completa del proceso de visualización involucrado en este prototipo puede verse en la Figura 7.7.

```

IF (hardware.GPU=TRUE)
  THEN
  IF (data.size == "LARGE")
    THEN
    Use Cubes for Nodes.
    Use Lines for Edges.
    Use Nothing for Visual Aids
    //the task is not considered in this case
    ELSE
    IF (hardware.monitor_size ="LARGE")
      THEN
      Use Spheres for Nodes.
      Use Cylinders for Edges.
      Use Nothing for Visual Aids.
      IF (isTaskFilter)
        THEN
        Use Triangles as Visual Aids only for the filtered elements.
        END
      ELSE
      IF (hardware.monitor.size = "REGULAR")
        THEN
        Use Cubes for Nodes.
        Use Lines for Edges.
        ELSE //hardware.monitor.size = "SMALL"
        Use Nothing for Nodes.
        Use Lines for Edges.
        END
      IF (isTaskFilter)
        THEN
        Use Triangles as Visual Aid only for the filtered elements.
        ELSE
        Use Triangles as Visual Aid for all the elements.
        END
      END
    END
  ELSE //hardware.GPU == FALSE
  Use Points for Nodes.
  Use Nothing for Lines.
  Use Nothing for Visual Aid.
  IF (isTaskFilter)
    THEN
    Use Triangles as Visual Aid only for the filtered elements.
    END
  END
END

```

Figura 7.6: Conjunto de reglas que relacionan la semántica de los datos, la tarea y el hardware para determinar cómo crear la visualización.

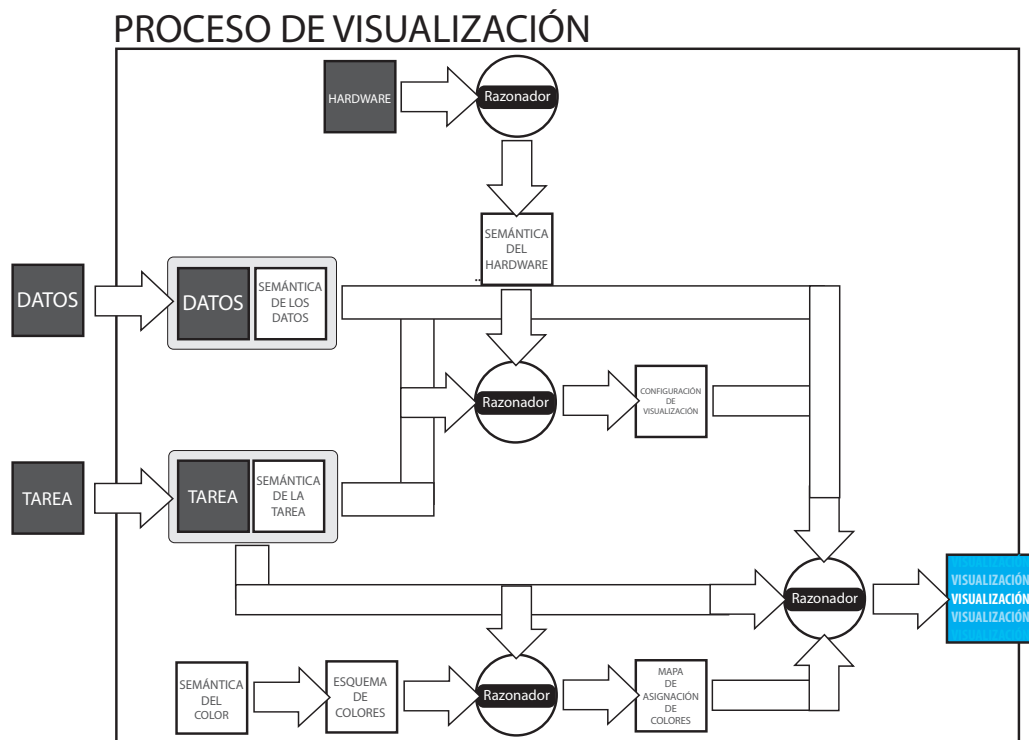


Figura 7.7: Arquitectura del sistema.

Capítulo 8

Conclusiones

Gracias a Internet y a la popularidad de las redes de computadoras, durante la última década se ha visto un crecimiento exponencial en el volumen de datos que es accedido y manipulado por los usuarios. No sólo se puede apreciar un crecimiento en el volumen de datos sino en la complejidad de los mismos lo que requiere de los usuarios mayores esfuerzos en identificar y comprender las relaciones relevantes para una tarea analítica particular. Mientras que el poder computacional ha acompañado este crecimiento, la habilidad de los usuarios para interactuar con los datos sólo se ha incrementado en forma lineal. La Visualización y en particular la Visualización de Información han resultado ser una herramienta fundamental para la comprensión de dichos datos.

Tal como se definió previamente, una visualización tiene como objetivo ampliar la capacidad cognitiva del usuario sobre un conjunto de datos. Una visualización que logra tal objetivo se reconoce como efectiva. Hoy en día, usuarios de distintas partes del mundo, en distintos dominios de aplicación y con diferentes conocimientos en sistemas de información requieren de visualizaciones efectivas para su trabajo diario. Gracias al aumento exponencial en el poder computacional y al abaratamiento de los costos de hardware, muchos de estos usuarios tienen acceso a una computadora con gran capacidad de procesamiento gráfico y a *software* para la creación de visualizaciones.

Sin duda contar con una posibilidad tan amplia de crear visualizaciones en 2D, 3D o nD es una gran ventaja, en especial si se compara con el poder computacional y la divulgación de la computación que existía hace 10 o 20 años atrás. Sin embargo, aún hoy en día no podemos asegurar que cada usuario esté creando la visualización más efectiva o siquiera una buena visualización para resolver su problema específico. La Visualización de

Información es un área claramente interdisciplinaria, tomando elementos de la psicología, el diseño gráfico, las ciencias de la computación, la ingeniería, la sociología y otras. Para poder asegurar la creación de una visualización son necesarios conocimientos no sólo en las áreas de visualización específicas sino también en las áreas previamente citadas. Este último condicionamiento implica que cada visualización, cada proceso de creación de una visualización requiere de conocimientos generales en el área de Visualización de Información, y de conocimientos particulares en el área de aplicación como puede ser la medicina, la ingeniería civil o la geología. Es muy improbable considerar que cada persona pueda ser capacitada para poder crear visualizaciones efectivas en su dominio de aplicación.

A partir de la problemática antes descrita surge nuestra propuesta de creación de un modelo de visualización que incorpore el conocimiento necesario del área de Visualización de Información para asegurar una visualización efectiva, y que posea además la flexibilidad necesaria para incorporar el conocimiento necesario de cada dominio de aplicación. De esta forma, sin importar el grado de conocimiento que tenga el usuario al momento de crear una representación visual, siempre se podrá asegurar una visualización efectiva como resultado. Para lograr este objetivo hemos utilizado la noción de semántica y las herramientas que ha brindado la Web Semántica. A partir de esto, en esta tesis se han propuesto:

- La definición de un modelo de visualización basado en semántica
- Una caracterización/ontología de los datos a visualizar.
- Una caracterización/ontología de las tareas a realizar sobre una visualización
- Una caracterización/ontología del humano. Aquí nos enfocamos tanto en la descripción del humano a partir de su información personal (*nombre, edad, lugar de nacimiento*) como también desde el punto de vista de HCI, considerando el modelo de procesamiento de información. Un tercer ítem que se incluye como parte de este trabajo es la caracterización del modelo cultural asociado al usuario.
- Una caracterización del contexto en el que ocurre la visualización. Por contexto consideramos el ambiente físico en el cual está trabajando el humano (*nivel de luz, de ruido*), el hardware que utiliza (*tipo de cpu, tipo de gpu, tamaño de monitor*), el contexto como dominio de aplicación en el que se encuentra la visualización (*ingeniería eléctrica, medicina, geología*) y también el dominio organizacional.

Gracias a los resultados obtenidos, hemos mostrado que mediante la incorporación de semántica es posible incorporar el conocimiento del área de Visualización en general en forma de semántica, la cual fue representada como ontologías. Cómo combinar apropiadamente colores o cómo configurar las técnicas de representación en función de la capacidad del hardware actual. Gracias a que hemos utilizado herramientas de la Web Semántica, nuestro sistema se basa en estándares abiertos, fácilmente extensibles y escalables. El uso de semántica como representación de conocimiento no sólo nos ha permitido crear un modelo de visualización que guía al usuario en la selección de los parámetros de la visualización sino también nos ha dado la posibilidad de derivar nueva información. Utilizando un razonador semántico hemos podido tomar el conocimiento incorporado al modelo de visualización y derivar nueva información. Un razonador semántico es capaz de derivar, a partir de la semántica de los datos y de la tarea a desarrollar, el mapa de color más efectivo para tal objetivo o determinar, conociendo el modelo cultural del usuario, qué diagramado de información será más apropiado.

El desarrollo de esta área de investigación aporta al desarrollo teórico básico, sumamente necesario en el campo emergente de la Visualización. Debido a la naturaleza interdisciplinaria del área de Visualización, los resultados obtenidos redundan en beneficios para diversos dominios de aplicación. La importancia de este trabajo de investigación radica en los resultados obtenidos que contribuirán al desarrollo teórico de base en Visualización, enriqueciendo el área de Visualización en general y el de Visualización Basada en Semántica en particular. Otro aporte de este trabajo es la contribución a la validación y refinamiento de los modelos de visualización desarrollados en el marco del VyGLab.

El desarrollo de esta tesis dio lugar a múltiples publicaciones: [LMC10b], [LC10], [LMC10a], [LMC07a], [LMC08], [LMC10c]. Este trabajo es el punto de inicio para múltiples líneas de investigación; tales líneas y el trabajo a futuro se detallan en el próximo capítulo.

Capítulo 9

Trabajo Futuro

El trabajo realizado en esta tesis podrá continuar durante los próximos años en diferentes direcciones. En este capítulo presentaremos brevemente los diferentes trabajos que se tienen planeado realizar a futuro como continuación del trabajo presentado aquí. Algunos de estos temas serán continuados por el autor de esta tesis mientras que otros serán desarrollados por nuevos tesistas como parte de trabajos de sus tesis de doctorados y magister. Todos los resultados obtenidos serán publicados en medios de difusión científica internacionales.

9.1. Validación de los resultados

En el desarrollo de esta tesis se mostró que es posible asistir al usuario en la creación de visualizaciones efectivas; queda aún por determinar el grado de efectividad de dichos resultados. Es necesario definir métricas de efectividad y diseñar las evaluaciones necesarias para su medición. Dichas evaluaciones no sólo permitirán reafirmar la validez de los resultados sino también serán de utilidad para alimentar la semántica definida. Debemos aún comparar los resultados obtenidos con aquéllos de otros modelos o aplicaciones existentes para la creación de visualizaciones.

9.2. Prototipado

Actualmente se está desarrollando un prototipo que busca incorporar a Brows.AR la ontología de datos, contexto, modelo cultural y valoración. Posteriormente se ha planteado el desarrollo de un prototipo que integrará el trabajo de Sebastián Escarza con los resultados de esta tesis. Esto consiste en definir aquellos elementos que sirvan de puente entre su Ontología de Visualización y la capa de inferencia. Es nuestro objetivo continuar trabajando con la aplicación Brows.AR como base para el testeo de la validez de nuestros resultados.

9.3. Líneas de investigación

A partir del trabajo realizado surgen varias líneas de investigación que abarcan, desde una profundización en las distintas caracterizaciones hasta la definición de ontologías de visualización; muchos de estos temas sólo representan un punto inicial de desarrollo. A continuación se mencionan las líneas de trabajo más relevantes:

- **Semántica del Usuario.** Se debe continuar investigando qué información incorporar a la semántica del usuario. Para esto se debe trabajar en extender la semántica definida para incorporar no sólo el modelo cultural sino también las habilidades físicas y cognitivas del usuario. En la realización de este trabajo surge la necesidad de incorporar al grupo a profesionales especializados que puedan llevar a cabo evaluaciones de dichas habilidades.
- **Semántica de las Tareas.** Toda visualización se crea con un objetivo, ya sea explorar, comparar o buscar patrones en los datos. Se debe continuar extendiendo la lista de tareas soportadas y la semántica asociada. Queda aún por mostrar que la clasificación de las tareas sea apropiada para cualquier dominio de aplicación.
- **Semántica del Contexto.** Al analizar el “contexto” lo que surge como relevante es el *hardware*. Sin embargo, el “contexto” está constituido por el *hardware*, el ambiente, el dominio de aplicación y el dominio organizacional. El *hardware* de las computadoras es altamente cambiante. Se crean dispositivos de salida de gran tamaño, como monitores o proyectores y a su vez también de pequeño tamaño,

como los dispositivos móviles. La semántica del contexto se debe adaptar a estos cambios para incorporar la semántica de los nuevos productos. Se debe continuar investigando qué elementos del ambiente físico afectan al usuario, a la visualización y al proceso de creación e incorporar dichos elementos y su semántica asociada.

- **Semántica de las Técnicas.** Las técnicas de representación transforman datos en formas visibles que resaltan aspectos importantes, como anomalías y similitudes en los datos. Estas representaciones visuales logran que el usuario perciba, en forma rápida, aspectos significativos de su conjunto de datos. Aumentando la capacidad cognitiva del usuario a través de representaciones visuales se logra un proceso analítico más rápido y enfocado. Sin duda es un gran reto elegir representaciones visuales adecuadas. Se debe definir una semántica uniforme que permita describir todas las técnicas de representación existentes y que posea la flexibilidad necesaria para incorporar nuevas técnicas.
- **Semántica de los Elementos Visuales.** A partir de los elementos visuales definidos por Sebastián Escarza en su tesis, se debe definir la semántica asociada e incluirla en el proceso de visualización basado en semántica.
- **Semántica de los Colores.** Existen diferentes representaciones de colores, tales como RGB, CMY, HSL o HSV. De acuerdo al tipo de tarea que se realiza sobre los colores algunos sistemas de representación son mejores que otros. Es necesario definir una semántica uniforme que incorpore todos los sistemas actuales de representación y que permita un fácil intercambio entre ellos. Es decir, si se cuenta con una representación visual que utiliza el sistema RGB se debería poder cambiar fácilmente a HSL si esto es necesario.

Apéndice A

Combinación de Colores

En el capítulo 7 se introdujo la implementación de dos ontologías, *Color* y *Colores*. La ontología *Color* sirve de descripción del concepto *Color* y sus relaciones, mientras que la ontología *Colores* contiene las instancias del concepto *Color*.

Dentro de la ontología *Color* se incluyeron las relaciones *opposite* y *next*, las cuales fueron luego usadas como parte de la descripción de la tarea *filter*. Las combinaciones de colores que se utilizaron en la ontología *Colores* se basó en la rueda de colores (Figura A.1). Estas combinaciones fueron validadas mediante una evaluación con usuarios que se realizó en la Universidad Nacional del Sur. Este anexo tiene como objetivo describir dicha evaluación.

A.1. Evaluación con Usuarios

La evaluación que se realizó con los usuarios fue llamada “Evaluación de Combinación de Colores” y tuvo como objetivo evaluar cómo el usuario percibe los colores cuando son combinados entre sí.

A.1.1. Técnica de Evaluación

La técnica utilizada para esta evaluación fue la de rellenado de un formulario. A los usuarios se les presentó una serie de tarjetas, a través de una presentación en una computadora, en las cuales se mostraban tres colores. Cada tarjeta estaba compuesta de

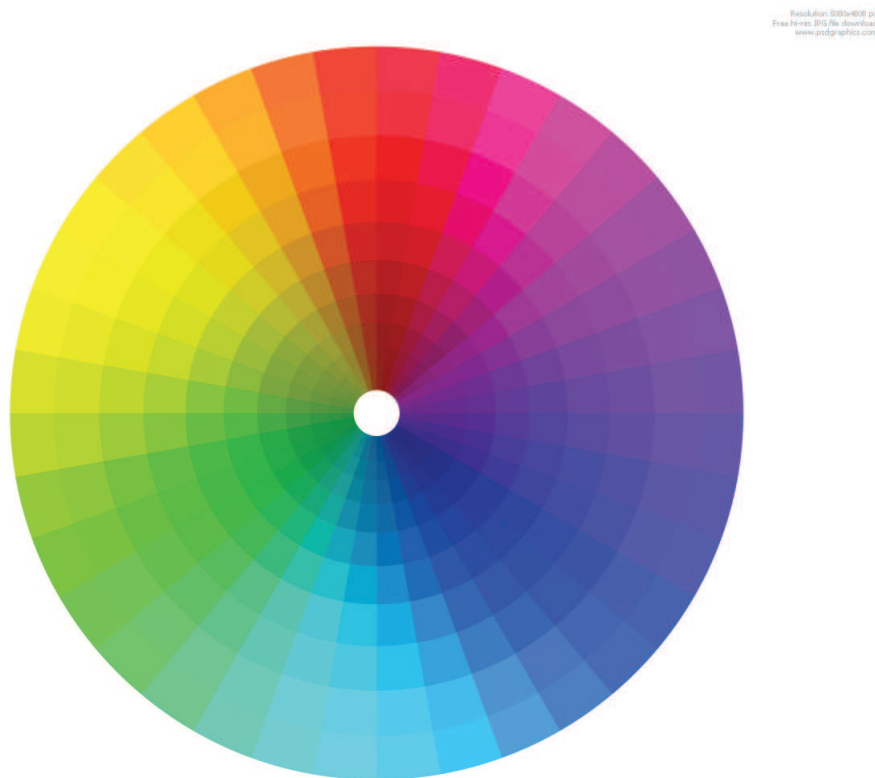


Figura A.1: Rueda de colores discretizada.

un rectángulo, a modo de fondo de la tarjeta, y dos círculos ubicados en paralelo en el centro de la misma (Figura A.2). El color utilizado para rellenar el rectángulo corresponde al concepto de *Background* presentado en la ontología de *Color*. Uno de los círculos fue rellenado con el color asociado a *Highlight* y el otro con el color asociado a *Regular*.

A cada usuario se le entregó un formulario con ocho tarjetas en blanco, tal como se ve en la Figura A.3 y se le pidió que, por cada tarjeta que se le mostraba, marcara qué círculo era el que se destacaba en la tarjeta. Las 8 tarjetas presentadas al usuario son las que se muestran en la Figura A.4.

A.1.2. Objetivos

En la sección 7 se presentaron dos relaciones entre colores, opuesto y semejante, las cuales se basaron en la rueda de colores (Figura A.1). El opuesto a un color se

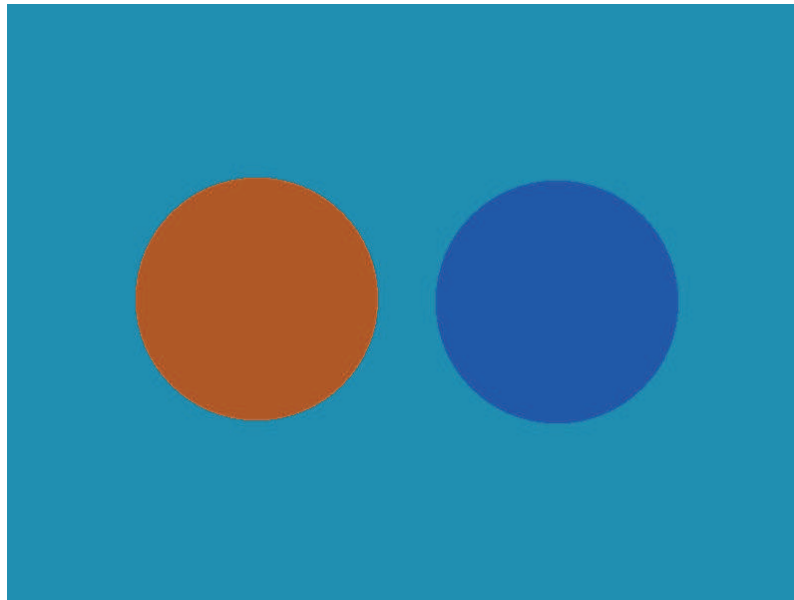


Figura A.2: Una de las tarjetas utilizadas en la “Evaluación de Combinación de Colores”.

describió como el opuesto en la rueda de colores y el semejante como uno de los colores vecinos o próximos. Esta evaluación tiene como objetivo verificar si dicha relación es verdad.

A.1.3. Usuarios y Lugar

Para llevar a cabo esta evaluación se convocaron 22 personas. Todos los involucrados fueron docentes de la Universidad Nacional del Sur; todos tenían experiencia en el uso de computadoras y ninguno presentaba problemas en la percepción de color. La evaluación tuvo lugar en los laboratorios del Departamento de Ciencias e Ingeniería de la Computación. Cada persona fue ubicada frente a su propia computadora. El objetivo de esta evaluación sólo fue explicado minutos antes de comenzar. Una vez ubicadas las personas en el laboratorio se procedió a explicar el objetivo de esta evaluación y la modalidad a utilizar. Luego de esto y al no presentarse dudas por parte de los convocados se inició la evaluación.

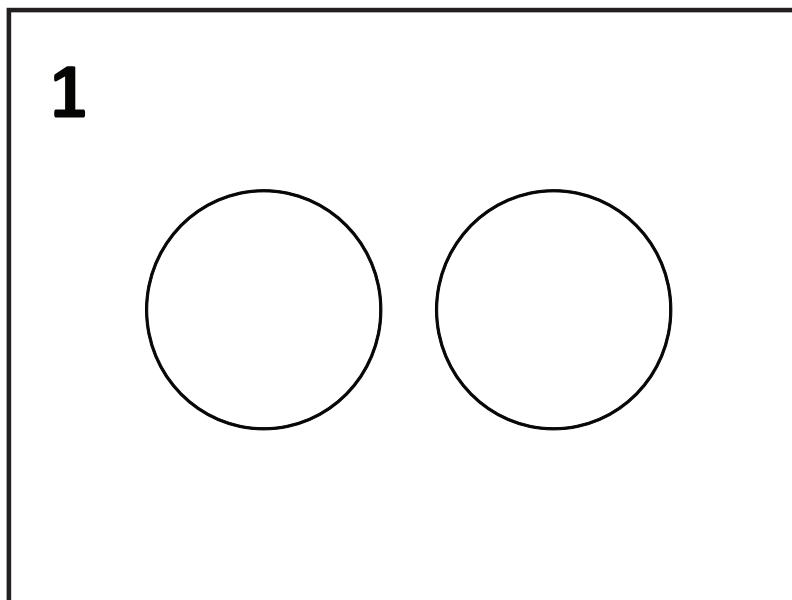


Figura A.3: Tarjetas en blanco a ser completada por el usuario. El formulario entregado a cada usuario estaba compuesto por 8 de estas tarjetas.

A.1.4. Resultados

En la Figura A.5 se pueden observar los resultados obtenidos. El número dentro de cada círculo representa la cantidad de personas que seleccionaron ese color. Claramente se puede apreciar que en ciertas combinaciones de colores, es fácil detectar cuál es el color que resalta, sin embargo esto no ocurre en todas las tarjetas.

A partir de los resultados obtenidos, se observó que los valores de luminancia y saturación en los colores usados variaban de un color a otro. Debido a esta observación se planteó la siguiente hipótesis: si los valores de luminancia y saturación son iguales para todos los colores, entonces se obtendrán mejores resultados. Para probar esta hipótesis se generaron nuevas tarjetas de colores (A.6), utilizando los mismos colores que en la evaluación anterior pero fijando la luminancia en 100 y la saturación en 200. Estos valores fueron elegidos ad hoc. La evaluación fue repetida con otras 22 personas y los nuevos resultados obtenidos mostraron ser mejores que los anteriores. En esta nueva evaluación se observó que más gente seleccionó el color opuesto al fondo como el color que resaltaba.

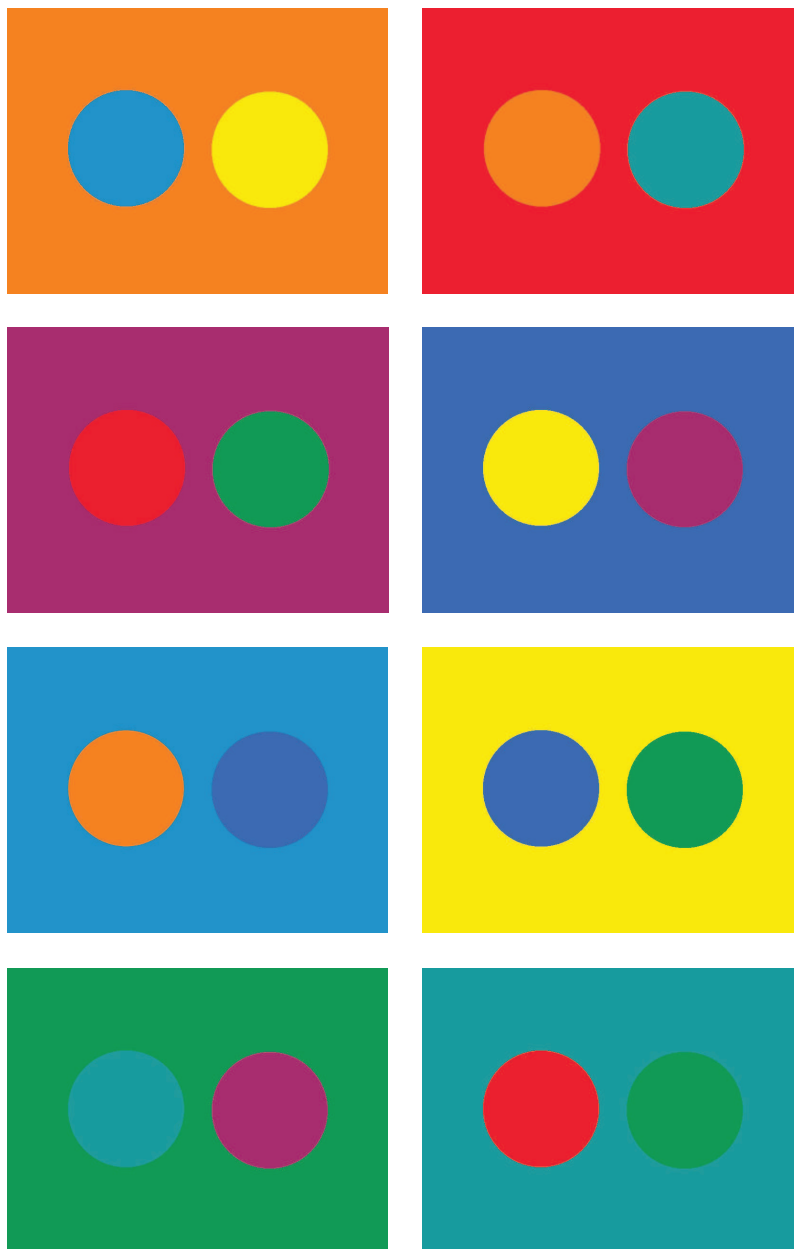


Figura A.4: Tarjeta usadas en la evaluación.

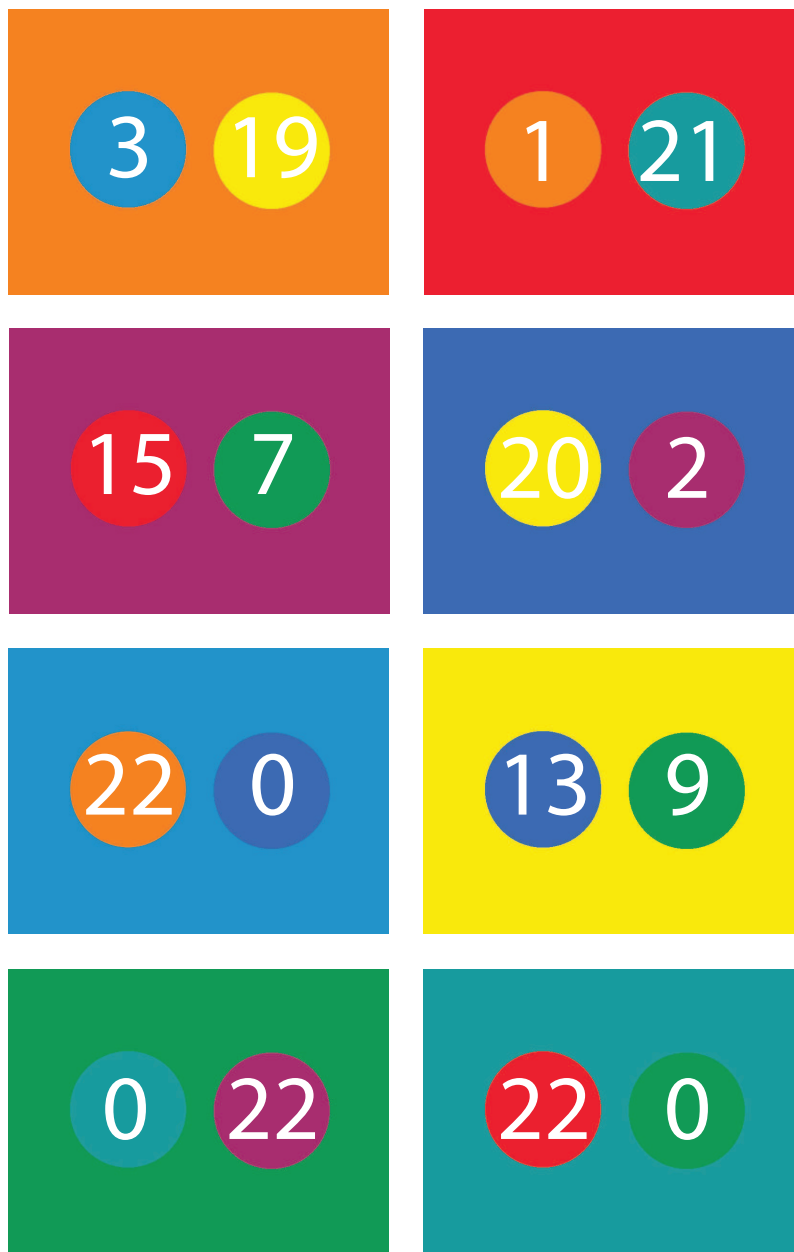


Figura A.5: Resultados de la evaluación. A los 22 participantes se les pidió que seleccionaran cuál era el color que resaltaba en cada tarjeta. El número dentro de cada círculo representa la cantidad de personas que seleccionaron ese color.

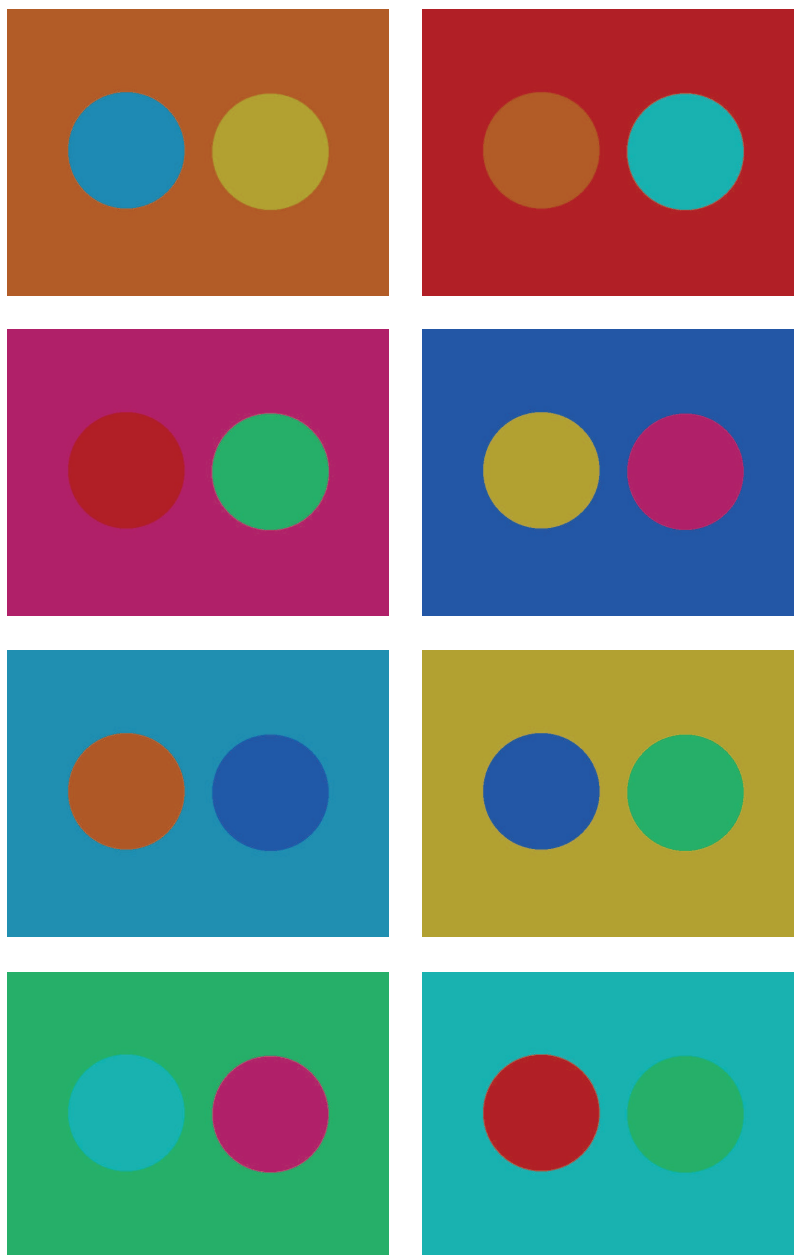


Figura A.6: Nuevas tarjetas usadas en la segunda iteración de la evaluación. Todos los colores poseen una luminancia de 100 y una saturación de 200.

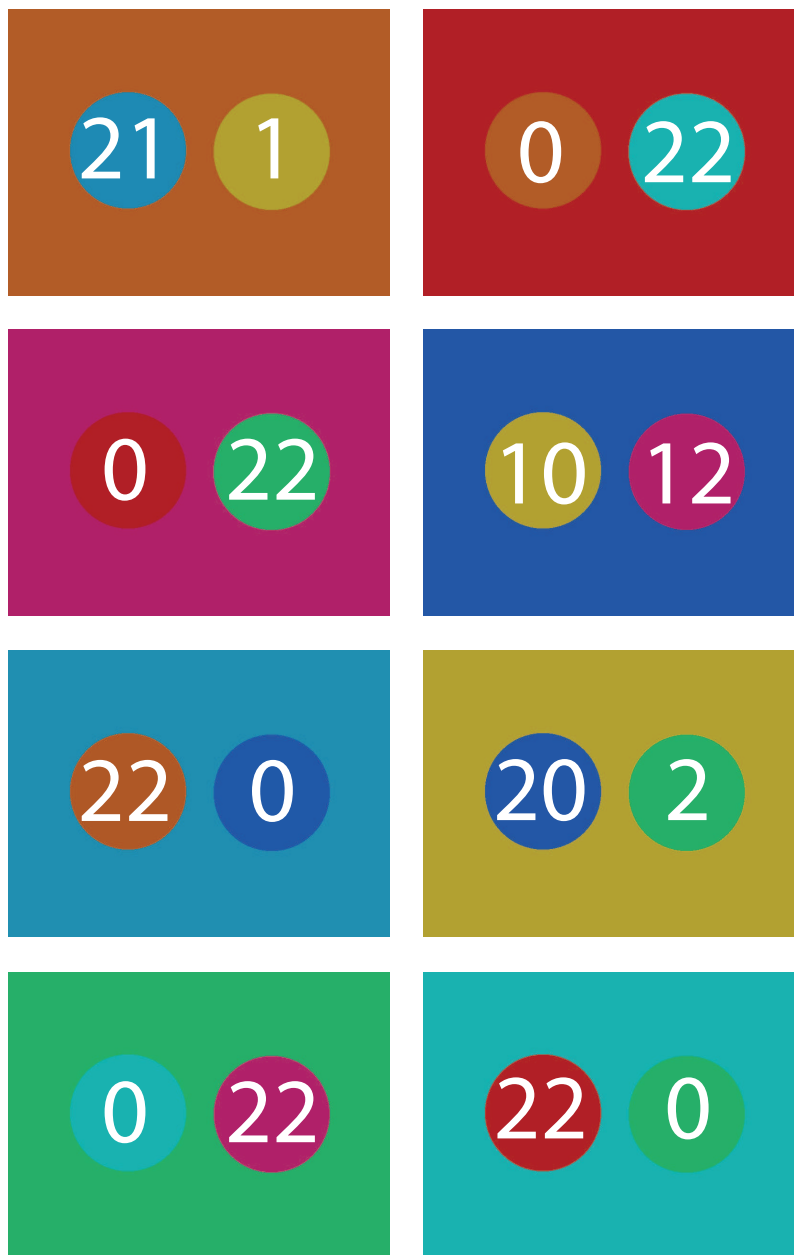


Figura A.7: Resultados de la evaluación con los valores normalizados. Los resultados demostraron ser mejores que los anteriores.

Bibliografía

- [BAB⁺93] BUTLER, D. M., ALMOND, J. C., BERGERON, R. D., BRODLIE, K. W., AND HABER, R. B. Visualization reference models. In *VIS '93: Proceedings of the 4th conference on Visualization '93* (Washington, DC, USA, 1993), IEEE Computer Society, pp. 337–342.
- [Bai96] BAILEY, R. W. *Human performance engineering (3rd ed.): designing high quality professional user interfaces for computer products, applications and systems*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.
- [BCE⁺92] BRODLIE, K. W., CARPENTER, L., EARNSHAW, R. A., GALLOP, J. R., HUBBOLD, R. J., MUMFORD, A. M., OSLAND, C. D., AND QUARENDON, P., Eds. *Scientific visualization: techniques and applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1992.
- [Ber83] BERTIN, J. *Semiology of graphics*. University of Wisconsin Press, 1983.
- [BGBG95] BAECKER, R. M., GRUDIN, J., BUXTON, W. A. S., AND GREENBERG, S., Eds. *Readings in Human Computer Interaction: Toward the Year 2000*, second ed. Morgan Kaufmann, 1995.
- [BGKS04] BOUZEGHOUB, M., GOBLE, C. A., KASHYAP, V., AND SPACCAPIETRA, S., Eds. *Semantic Visualization of Biochemical Databases* (2004), vol. 3226 of *Lecture Notes in Computer Science*, Springer.
- [BN07] BRODLIE, K., AND NOOR, N. M. Visualization notations, models and taxonomies. In *EG UK Theory and Practice of Computer Graphics* (2007), pp. 207–212.

- [BRT95] BERGMAN, L. D., ROGOWITZ, B. E., AND TREINISH, L. A. A rule-based tool for assisting colormap selection. In *VIS '95: Proceedings of the 6th conference on Visualization '95* (Washington, DC, USA, 1995), IEEE Computer Society, p. 118.
- [BRT09] BRESLOW, L. A., RATWANI, R. M., AND TRAFTON, J. G. Cognitive models of the influence of color scale on data visualization tasks. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 51, 3 (2009), 321–338.
- [BvHH⁺04] BECHHOFFER, S., VAN HARMELEN, F., HENDLER, J., HORROCKS, I., MCGUINNESS, D. L., PATEL-SCHNEIDER, P. F., AND STEIN, L. A. OWL Web Ontology Language Reference. Tech. rep., W3C, <http://www.w3.org/TR/owl-ref/>, February 2004.
- [Car03] CARPENDALE, M. S. T. Considering visual variables as a basis for information visualisation. Tech. rep., University of Calgary, Calgary, AB, 2003.
- [CDD⁺04] CARROLL, J. J., DICKINSON, I., DOLLIN, C., REYNOLDS, D., SEABORNE, A., AND WILKINSON, K. Jena: implementing the semantic web recommendations. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters* (New York, NY, USA, 2004), WWW Alt. '04, ACM, pp. 74–83.
- [CEH⁺09] CHEN, M., EBERT, D., HAGEN, H., LARAMEE, R. S., VAN LIERE, R., MA, K.-L., RIBARSKY, W., SCHEUERMANN, G., AND SILVER, D. Data, information, and knowledge in visualization. *IEEE Comput. Graph. Appl.* 29 (January 2009), 12–19.
- [CFLGP03] CORCHO, O., FERNÁNDEZ-LÓPEZ, M., AND GÓMEZ-PÉREZ, A. Methodologies, tools and languages for building ontologies: where is their meeting point? *Data Knowl. Eng.* 46 (July 2003), 41–64.
- [CGC07] CHALLAM, V., GAUCH, S., AND CHANDRAMOULI, A. Contextual search using ontology-based user profiles. In *RIAO* (2007).

- [Chi00] CHI, E. H. A taxonomy of visualization techniques using the data state reference model. In *INFOVIS '00: Proceedings of the IEEE Symposium on Information Visualization 2000* (Washington, DC, USA, 2000), IEEE Computer Society, p. 69.
- [Cle85] CLEVELAND, W. S. *The elements of graphing data*. Wadsworth Publ. Co., Belmont, CA, USA, 1985.
- [CM84] CLEVELAND, W. S., AND MCGILL, R. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association* 79, 387 (1984), 531–554.
- [CM97] CARD, S. K., AND MACKINLAY, J. The structure of the information visualization design space. In *INFOVIS '97: Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis '97)* (Washington, DC, USA, 1997), IEEE Computer Society, p. 92.
- [CMS99] CARD, S. K., MACKINLAY, J. D., AND SHNEIDERMAN, B., Eds. *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.
- [CR96] CHUAH, M. C., AND ROTH, S. F. On the semantics of interactive visualizations. In *INFOVIS '96: Proceedings of the 1996 IEEE Symposium on Information Visualization (INFOVIS '96)* (Washington, DC, USA, 1996), IEEE Computer Society, p. 29.
- [CR98] CHI, E. H.-H., AND RIEDL, J. An operator interaction framework for visualization systems. In *INFOVIS '98: Proceedings of the 1998 IEEE Symposium on Information Visualization* (Washington, DC, USA, 1998), IEEE Computer Society, pp. 63–70.
- [DBD04] DUKE, D. J., BRODLIE, K. W., AND DUCE, D. A. Building an ontology of visualization. In *VIS '04: Proceedings of the conference on Visualization '04* (Washington, DC, USA, 2004), IEEE Computer Society, p. 598.7.
- [DBDH05] DUKE, D. J., BRODLIE, K. W., DUCE, D. A., AND HERMAN, I. Do you see what i mean? *IEEE Comput. Graph. Appl.* 25, 3 (2005), 6–9.

- [DFvH03] DAVIES, J., FENSEL, D., AND VAN HARMELEN, F., Eds. *Towards the Semantic Web: Ontology-driven Knowledge Management*. Wiley, Chichester, UK, 2003.
- [Die05] DIESTEL, R. *Graph Theory*, electronic edition ed. Springer-Verlag, 2005.
- [ECM07] ESCARZA, S., CASTRO, S., AND MARTIG, S. Interaction on graphs using non-conventional devices. In *Journal of Computer Science and Technology* (Executive Offices. University of New Mexico - MSC 02-1600 Albuquerque, NM 87131-0001, USA, 2007), ISTECS, pp. 59–65.
- [ECSM06] ESCARZA, S., CASTRO, S., AND MARTIG, S. Interacciones sobre grafos mediante dispositivos no convencionales. In *Proceedings of the XII Congreso Argentino de Ciencias de la Computación* (2006), pp. 771–783.
- [Esc09a] ESCARZA, S. Vyglab-technical report-007: Implementación del muv: La ontología de visualización. Tech. rep., Universidad Nacional del Sur. Departamento de Ciencias e Ingeniería de la Computación. Laboratorio de Investigación y Desarrollo en Visualización y Computación Gráfica, Av. Alem 1253. Bahía Blanca. Buenos Aires. Argentina, 2009.
- [Esc09b] ESCARZA, S. Vyglab-technical report-008: Implementación del muv: Aplicando la ontología a un ejemplo. Tech. rep., Universidad Nacional del Sur. Departamento de Ciencias e Ingeniería de la Computación. Laboratorio de Investigación y Desarrollo en Visualización y Computación Gráfica, Av. Alem 1253. Bahía Blanca. Buenos Aires. Argentina, 2009.
- [Fle72] FLEISHMAN, E. A. On the relation between abilities, learning, and human performance. *American Psychologist* 27, 11 (1972), 1017 – 1032.
- [FMCL04] FABRITIUS, C. V., MADSEN, N. L., CLAUSEN, J., AND LARSEN, J. Finding the best visualization of an ontology. Tech. rep., Informatics and Mathematical Modelling, Technical University of Denmark, (DTU), Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2004.
- [FMK+08] FLOURIS, G., MANAKANATAS, D., KONDYLAKIS, H., PLEXOUSAKIS, D., AND ANTONIOU, G. Ontology change: Classification and survey. *Knowl. Eng. Rev.* 23 (June 2008), 117–152.

- [FP98] FENTON, N. E., AND PFLEEGER, S. L. *Software Metrics: A Rigorous and Practical Approach*, 2 ed. PWS Publishing Company, 1998.
- [Fur86] FURNAS, G. W. Generalized fisheye views. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 1986), CHI '86, ACM, pp. 16–23.
- [FvHH⁺01] FENSEL, D., VAN HARMELEN, F., HORROCKS, I., MCGUINNESS, D. L., AND PATEL-SCHNEIDER, P. F. Oil: An ontology infrastructure for the semantic web. *IEEE Intelligent Systems* 16 (2001), 38–45.
- [GC00] GRUDIN, J., AND CARROLL, J. M., Eds. *ACM Trans. Comput.-Hum. Interact.*, vol. 7. ACM, New York, NY, USA, 2000.
- [GHV⁺06] GOLEMATI, M., HALATSIS, C., VASSILAKIS, C., KATIFORI, A., AND OF PELOPONNESE, U. A context-based adaptive visualization environment. *Proceedings of the conference on Information Visualization* (2006), 62–67.
- [GKV⁺07] GOLEMATI, M., KATIFORI, A., VASSILAKIS, C., LEPOURAS, G., AND HALATSIS, C. Creating an ontology for the user profile: Method and applications. In *In Proceedings of the First International Conference on Research Challenges in Information Science (RCIS)* (2007).
- [GMF⁺03] GENNARI, J. H., MUSEN, M. A., FERGERSON, R. W., GROSSO, W. E., CRUBÉZY, M., ERIKSSON, H., NOY, N. F., AND TU, S. W. The evolution of protege: an environment for knowledge-based systems development. *Int. J. Hum.-Comput. Stud.* 58 (January 2003), 89–123.
- [Gro09] GROBE, M. Rdf, jena, sparql and the 'semantic web'. In *Proceedings of the 37th annual ACM SIGUCCS fall conference* (New York, NY, USA, 2009), SIGUCCS '09, ACM, pp. 131–138.
- [Gru93] GRUBER, T. R. A translation approach to portable ontology specifications. *KNOWLEDGE ACQUISITION* 5 (1993), 199–220.
- [HF01] HARMELEN, F. V., AND FLUIT, C. Ontology-based information visualisation. In *IV '01: Proceedings of the Fifth International Conference on Information Visualisation* (Washington, DC, USA, 2001), IEEE Computer Society, p. 546.

- [HM03] HAARSLEV, V., AND MÖLLER, R. Racer: An owl reasoning agent for the semantic web. In *Proc. Int'l Wkshp on Applications, Products and Services of Web-based Support Systems (Held at 2003 IEEE/WIC Int'l Conf. on Web Intelligence)* (2003), Society Press, pp. 91–95.
- [Hof03] HOFSTEDE, G., Ed. *Culture's Consequences: Comparing values, behaviours and organisations across nations*. Sage Publications Inc, Thousand Oaks, 2003.
- [Hor07] HORROCKS, I. Semantic web: the story so far. In *Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A)* (New York, NY, USA, 2007), W4A '07, ACM, pp. 120–125.
- [JKMG07] JANKUN-KELLY, T. J., MA, K.-L., AND GERTZ, M. A model and framework for visualization exploration. *IEEE Transactions on Visualization and Computer Graphics* 13, 2 (2007), 357–369.
- [JS03] JACKO, J. A., AND SEARS, A., Eds. *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 2003.
- [KCM06] KALOGERAKIS, E., CHRISTODOULAKIS, S., AND MOUMOUTZIS, N. Coupling ontologies with graphics content for knowledge driven visualization. In *VR '06: Proceedings of the IEEE conference on Virtual Reality* (Washington, DC, USA, 2006), IEEE Computer Society, pp. 43–50.
- [KFNM04] KNUBLAUCH, H., FERGERSON, R. W., NOY, N. F., AND MUSEN, M. A. The protégé owl plugin: An open development environment for semantic web applications. Springer, pp. 229–243.
- [KK94] KELLER, P. R., AND KELLER, M. M. *Visual Cues: Practical Data Visualization*. IEEE Computer Society Press, Los Alamitos, CA, USA, 1994.
- [KMR04] KNUBLAUCH, H., MUSEN, M. A., AND RECTOR, A. L. Editing description logic ontologies with the protégé owl plugin. In *International Workshop on Description Logics* (2004).

- [Koo08] KOOP, D. Viscomplete: Automating suggestions for visualization pipelines. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (2008), 1691–1698.
- [Kov04] KOVALERCHUK, BORIS; SCHWING, J. E. *Visual and Spatial Analysis*. Springer, 2004.
- [Lar06] LARREA, M. Thesis overview: Spherical layout for 3d graph visualization. In *Journal of Computer Science and Technology* (Executive Offices. University of New Mexico - MSC 02-1600 Albuquerque, NM 87131-0001, USA, 2006), ISTECS, pp. 112–113.
- [LC10] LARREA, M., AND CASTRO, S. Semantics-based visualization. In *Proceedings of the XVI Congreso Argentino de Ciencias de la Computación* (2010), U. de Moron, Ed., no. 1 in XVI Congreso Argentino de Ciencias de la Computación, pp. 1060–1065.
- [LMC07a] LARREA, M., MARTIG, S., AND CASTRO, S. Semantic based visualization. In *Proceedings of the IX Workshop de Investigadores en Ciencias de la Computación* (2007), U. N. de la Patagonia San Juan Bosco, Ed., no. 1 in IX Workshop de Investigadores en Ciencias de la Computación, pp. 289–293.
- [LMC07b] LARREA, M., MARTIG, S., AND CASTRO, S. Spherical layout - layout for 3d tree visualization. In *Proceedings of the IADIS 2007, Multi Conference on Computer Science and Information Systems* (2007), pp. 91–98.
- [LMC08] LARREA, M., MARTIG, S., AND CASTRO, S. Semantic based visualization: A first approach. In *Proceedings of the X Workshop de Investigadores en Ciencias de la Computación* (2008), U. N. de La Pampa, Ed., no. 1 in X Workshop de Investigadores en Ciencias de la Computación, pp. 125–131.
- [LMC10a] LARREA, M., MARTIG, S., AND CASTRO, S. Semantics-based color assignment in visualization. In *Journal of Computer Science and Technology* (Executive Offices. University of New Mexico - MSC 02-1600 Albuquerque, NM 87131-0001, USA, 2010), ISTECS, pp. 14–18.
- [LMC10b] LARREA, M., MARTIG, S., AND CASTRO, S. A semantics-based visualization building process. In *Proceedings of the XVI Congreso Argentino*

- de Ciencias de la Computación* (2010), U. de Moron, Ed., no. 1 in XVI Congreso Argentino de Ciencias de la Computación, pp. 465–474.
- [LMC10c] LARREA, M., MARTIG, S., AND CASTRO, S. Visualización basada en semántica. In *Proceedings of the XII Workshop de Investigadores en Ciencias de la Computación* (2010), U. N. de la Patagonia San Juan Bosco, Ed., no. 1 in XII Workshop de Investigadores en Ciencias de la Computación, pp. 201–208.
- [LPP+06] LEE, B., PLAISANT, C., PARR, C. S., FEKETE, J.-D., AND HENRY, N. Task taxonomy for graph visualization. In *BELIV '06: Proceedings of the 2006 AVI workshop on BEyond time and errors* (New York, NY, USA, 2006), ACM, pp. 1–5.
- [LUMC09] LARREA, M., URRIBARRI, D., MARTIG, S., AND CASTRO, S. Spherical layout implementation using centroidal voronoi tessellations. *CoRR abs/0912.3974* (2009). informal publication.
- [Mac86] MACKINLAY, J. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.* 5, 2 (1986), 110–141.
- [MC06] MARTIG, S., AND CASTRO, S. Interacciones básicas en el modelo unificado ed visualización. In *Proceedings of the XII Congreso Argentino de Ciencias de la Computación* (2006), pp. 747–760.
- [McB02] MCBRIDE, B. Jena: A semantic web toolkit. *IEEE Internet Computing* 6 (November 2002), 55–59.
- [MCFE03] MARTIG, S., CASTRO, S., FILLOTTRANI, P., AND ESTÉVEZ, E. Un modelo unificado de visualización. In *Proceedings 9º Congreso Argentino de Ciencias de la Computación* (2003), pp. 881–892.
- [MCGS08] MOTIK, B., CUENCA GRAU, B., AND SATTTLER, U. Structured objects in owl: representation and reasoning. In *Proceeding of the 17th international conference on World Wide Web* (New York, NY, USA, 2008), WWW '08, ACM, pp. 555–564.
- [MFHS02] MCGUINNESS, D. L., FIKES, R., HENDLER, J., AND STEIN, L. A. Daml+oil: An ontology language for the semantic web. *IEEE Intelligent Systems* 17 (September 2002), 72–80.

- [NHN02] NGUYEN, T. D., HO, T. B., AND NGUYEN, D. Data and knowledge visualization in knowledge discovery process. In *VISUAL '02: Proceedings of the 5th International Conference on Recent Advances in Visual Information Systems* (London, UK, 2002), Springer-Verlag, pp. 311–322.
- [PHP03] PFITZNER, D., HOBBS, V., AND POWERS, D. A unified taxonomic framework for information visualization. In *APVis '03: Proceedings of the Asia-Pacific symposium on Information visualisation* (Darlinghurst, Australia, Australia, 2003), Australian Computer Society, Inc., pp. 57–66.
- [PRH⁺06] PULIDO, J. R. G., RUIZ, M. A. G., HERRERA, R., CABELLO, E., LEGRAND, S., AND ELLIMAN, D. Ontology languages for the semantic web: A never completely updated review. *Know.-Based Syst. 19* (November 2006), 489–497.
- [PSHH04] PATEL-SCHNEIDER, P. F., HAYES, P., AND HORROCKS, I. Owl web ontology language semantics and abstract syntax. Tech. rep., W3C, 2004.
- [QZP03] QIN, C., ZHOU, C., AND PEI, T. Taxonomy of visualization techniques and systems - concerns between users and developers are different. In *Asia GIS Conference 2003* (2003).
- [RB09] REINECKE, K., AND BERNSTEIN, A. Tell me where you've lived, and i'll tell you what you like: Adapting interfaces to cultural preferences. In *Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization: formerly UM and AH* (Berlin, Heidelberg, 2009), UMAP '09, Springer-Verlag, pp. 185–196.
- [RBG07] RAUTEK, P., BRUCKNER, S., AND GROLLER, E. Semantic layers for illustrative volume rendering. *IEEE Transactions on Visualization and Computer Graphics 13* (November 2007), 1336–1343.
- [Rei07] REINECKE, K. Cultural adaptivity for the semantic web. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference* (Berlin, Heidelberg, 2007), ISWC'07/ASWC'07, Springer-Verlag, pp. 946–950.

- [Ren94] RENNISON, E. Galaxy of news: an approach to visualizing and understanding expansive news landscapes. In *UIST '94: Proceedings of the 7th annual ACM symposium on User interface software and technology* (New York, NY, USA, 1994), ACM Press, pp. 3–12.
- [RK07] ROBAL, T., AND KALJA, A. Applying user profile ontology for mining web site adaptation recommendations. In *ADBIS Research Communications* (2007), Y. E. Ioannidis, B. Novikov, and B. Rachev, Eds., vol. 325 of *CEUR Workshop Proceedings*, CEUR-WS.org.
- [RKMG94] ROTH, S. F., KOLOJEJCHICK, J., MATTIS, J., AND GOLDSTEIN, J. Interactive graphic design using automatic presentation knowledge. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 1994), ACM, pp. 112–117.
- [RM90] ROTH, S. F., AND MATTIS, J. Data characterization for intelligent graphics presentation. In *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 1990), ACM, pp. 193–200.
- [RM91] ROTH, S. F., AND MATTIS, J. Automating the presentation of information. In *Proceedings of the IEEE Conference on Artificial Intelligence Applications* (February 1991), pp. 90–97.
- [RRB07] REINECKE, K., REIF, G., AND BERNSTEIN, A. Cultural User Modeling With CUMO: An Approach to Overcome the Personalization Bootstrapping Problem. In *First International Workshop on Cultural Heritage on the Semantic Web at the 6th International Semantic Web Conference (ISWC 2007)* (Busan, South Korea, November 12 2007).
- [SAR08] SHU, G., AVIS, N. J., AND RANA, O. F. Bringing semantics to visualization services. *Adv. Eng. Softw.* 39, 6 (2008), 514–520.
- [SBI85] SCHMOLZE, J. G., BERANEK, B., AND INC, N. An overview of the kl-one knowledge representation system. *Cognitive Science* 9 (1985), 171–216.

- [SEZJM08] STAN, J., EGYED-ZSIGMOND, E., JOLY, A., AND MARET, P. A user profile ontology for situation-aware social networking. *3rd Workshop on Artificial Intelligence Techniques for Ambient Intelligence 1* (July 2008), 1–4.
- [SH00] STOLTE, C., AND HANRAHAN, P. Polaris: A system for query, analysis and visualization of multi-dimensional relational databases. In *INFOVIS '00: Proceedings of the IEEE Symposium on Information Visualization 2000* (Washington, DC, USA, 2000), IEEE Computer Society, p. 5.
- [Shn96] SHNEIDERMAN, B. The eyes have it: A task by data type taxonomy for information visualizations. In *VL '96: Proceedings of the 1996 IEEE Symposium on Visual Languages* (Washington, DC, USA, 1996), IEEE Computer Society, p. 336.
- [Shn97] SHNEIDERMAN, B. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997.
- [SML04] SCHROEDER, W., MARTIN, K., AND LORENSEN, B. *The Visualization Toolkit, Third Edition*. Kitware Inc., 2004.
- [SPG⁺07a] SIRIN, E., PARSIA, B., GRAU, B. C., KALYANPUR, A., AND KATZ, Y. Pellet: A practical owl-dl reasoner. *Web Semant. 5* (June 2007), 51–53.
- [SPG⁺07b] SIRIN, E., PARSIA, B., GRAU, B. C., KALYANPUR, A., AND KATZ, Y. Pellet: A practical owl-dl reasoner. *Web Semant. 5* (June 2007), 51–53.
- [Ste46] STEVENS, S. S. On the theory of scales of measurement. *Science 103*, 2684 (June 1946), 677–680.
- [STH02] STOLTE, C., TANG, D., AND HANRAHAN, P. Query, analysis, and visualization of hierarchically structured data using polaris. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2002), KDD '02, ACM, pp. 112–122.
- [STH08] STOLTE, C., TANG, D., AND HANRAHAN, P. Polaris: a system for query, analysis, and visualization of multidimensional databases. *Commun. ACM 51* (November 2008), 75–84.

- [SVK07] SCHEIDEGGER, C., VO, H., AND KOOP, D. Querying and creating visualizations by analogy. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1560–1567. Member-Freire, Juliana and Member-Silva, Claudio.
- [TH06] TSARKOV, D., AND HORROCKS, I. Fact++ description logic reasoner: System description. In *Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2006)* (2006), vol. 4130 of *Lecture Notes in Artificial Intelligence*, Springer, pp. 292–297.
- [TM02] TORY, M., AND MÖLLER, T. A model-based visualization taxonomy. Technical Report SFU-CMPT-TR2002-06, Computing Science Dept., Simon Fraser Univ., 2002.
- [TM04] TORY, M., AND MOLLER, T. Rethinking visualization: A high-level taxonomy. In *Proceedings of the IEEE Symposium on Information Visualization* (2004), pp. 151–158.
- [Tuf86] TUFTE, E. R. *The Visual Display of Quantitative Information*. Graphics Press, May 1986.
- [Twe97] TWEEDIE, L. Characterizing interactive externalizations. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 1997), ACM, pp. 375–382.
- [UCM06] URRIBARRI, D., CASTRO, S., AND MARTIG, S. Interacciones sobre coordenadas paralelas en el marco del modelo unificado ed visualización. In *Proceedings of the XII Congreso Argentino de Ciencias de la Computación* (2006), pp. 761–770.
- [UFK⁺89] UPSON, C., FAULHABER, T. A., KAMINS, D., LAIDLAW, D., SCHLEGEL, D., VROOM, J., GURWITZ, R., AND VAN DAM, A. The application visualization system: a computational environment for scientific visualization. *Computer Graphics and Applications, IEEE* 9, 4 (1989), 30–42.
- [VHS07] VERMA, R., HANSCH, C., AND SELASSIE, C. Comparative qsar studies on pampa/modified pampa for high throughput profiling of drug absorption potential with respect to caco-2 cells and human intestinal

- absorption. *Journal of Computer-Aided Molecular Design* 21 (2007), 3–22. 10.1007/s10822-006-9101-z.
- [WB98] WENG, Z. M., AND BELL, D. Integrating visual ontologies and wavelets for image content retrieval. In *DEXA '98: Proceedings of the 9th International Workshop on Database and Expert Systems Applications* (Washington, DC, USA, 1998), IEEE Computer Society, p. 379.
- [WL90] WEHREND, S., AND LEWIS, C. A problem-oriented classification of visualization techniques. In *VIS '90: Proceedings of the 1st conference on Visualization '90* (Los Alamitos, CA, USA, 1990), IEEE Computer Society Press, pp. 139–143.
- [Woo85] WOODS, W. A. What's in a link: Foundations for semantic networks. In *Readings in Knowledge Representation*, R. J. Brachman and H. J. Levesque, Eds. Kaufmann, Los Altos, CA, 1985, pp. 217–241.
- [XCW05] XU, Z., CHEN, H., AND WU, Z. Applying semantic web technologies for geodata integration and visualization. In *ER (Workshops)* (2005), pp. 320–329.
- [YZ92] YAGER, R. R., AND ZADEH, L. A., Eds. *An Introduction to Fuzzy Logic Applications in Intelligent Systems*. Kluwer Academic Publishers, Norwell, MA, USA, 1992.
- [ZF96] ZHOU, M. X., AND FEINER, S. K. Data characterization for automatically visualizing heterogeneous information. In *INFOVIS '96: Proceedings of the 1996 IEEE Symposium on Information Visualization (INFOVIS '96)* (Washington, DC, USA, 1996), IEEE Computer Society, p. 13.
- [Zha96] ZHANG, J. A representational analysis of relational information displays. *Int. J. Hum.-Comput. Stud.* 45, 1 (1996), 59–74.