DISS. ETH NO. 15388

INFOTICLES: INFORMATION VISUALIZATION USING DATA-DRIVEN PARTICLES

A dissertation submitted to the

SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH

for the degree of

Doctor of Technical Sciences

presented by

ANDREW ELZAR VANDE MOERE

Ir.Arch., Katholieke Universiteit Leuven

born 18.03.1975

citizen of Belgium

accepted on the recommendation of

Prof. Dr. Gerhard Schmitt, examiner
Prof. Dr. Markus Gross, co-examiner
Dr. Chris Luebkeman, co-examiner

2004

Abstract

Insights from information architecture, computer graphics and cognitive science are merged to create the concept of data-driven particles, coined *infoticles*, which simulate the evolution of abstract data in time. Conceptually, each infoticle represents a unique data object, retrieved from a database. The spatial behavior of an infoticle is determined by two data-driven principles: external influences that follow the rules of Newtonian mechanics, and internal interdependencies based upon self-organizing rules. Infoticles convey time-varying information as a result of the automatic generation of cognitively interpretable motion typologies and the creation of static artifacts with distinct visual features that trace these dynamic behaviors.

Data-driven particles are a novel information visualization metaphor for exploring large, time-varying datasets, which typically consist of reoccurring data objects that have been altered in time. This visualization technique is able to effectively represent various dynamic data characteristics, such as relative data value changes and update frequencies, for each individual data entry or in the global context of the whole dataset.

The infoticle metaphor merges the qualities of information-aesthetics and user-engagement with the unique characteristics of immersive virtual reality, such as stereoscopic vision and spatial awareness. The infoticle methodology describes in detail the adopted design rationale in the context of scientific insights originating from diverse related disciplines, ranging from human cognition over information architecture to the field of computer graphics.

Most system implementation issues are generally valid for information visualization applications that query, communicate, process, update and simulate continuous, large streams of time-varying datasets in real time. Four different prototypes that apply time-varying datasets retrieved from the worlds of finance, stock market and knowledge management demonstrate the versatility of the proposed visualization method. A subsequent overall analysis reveals the grammatical principles and syntax of the emergent visual patterns, and investigates the various qualities and potential limitations of the proposed methodology.

This research was accomplished within the framework of the blue-c project at ETH-Zurich, which developed a novel virtual reality environment with tele-immersive capabilities.

Zusammenfassung

Erkenntnisse aus der Informationsarchitektur, Computergrafik und den kognitiven Wissenschaften sind die Grundlage für das Konzept datengesteuerter Partikel, genannt Infoticles, welche die Entwicklung abstrakter Daten in Abhängigkeit von Zeit simulieren. Auf konzeptueller Ebene repräsentiert jedes Infoticle ein einzelnes Datenobjekt, das von einer Datenbank angefordert wurde. Das räumliche Verhalten eines Infoticles wird durch zwei datengesteuerte Grundprinzipien bestimmt: Externe Einflüsse, die den Gesetzen Newtonscher Mechanik folgen und interne Abhängigkeiten, die auf selbstorganisierenden Regeln basieren. Infoticles vermitteln zeitabhängige Informationen durch die Verwendung automatisch generierter, interpretierbarer Bewegungstypologien und durch das Erschaffen statischer Artefakte mit unterschiedlichen visuellen Merkmalen, die das dynamische Verhalten sichtbar machen.

Datengesteuerte Partikel sind eine neue Informationsvisualisierungsmetapher zum Untersuchen grosser, zeitabhängiger Datensätze, welche oft aus gleichen Datenobjekten bestehen, die sich über die Zeit verändern. Die Visualisierungsmethode erlaubt es, verschiedene dynamische Eigenschaften von Daten effektiv darzustellen; beispielsweise relative Veränderungen von Datenwerten oder Aktualisierungsfrequenzen im Kontext des gesamten Datensatzes.

Die Infoticle-Metapher verbindet die Qualitäten von Informationsästhetik und Benutzerengagement mit den spezifischen Möglichkeiten immersiver virtueller Realität, wie stereoskopischer Sicht oder räumlicher Wahrnehmung. Die Infoticle-Methode beschreibt detailliert das verfolgte Designkonzept und erklärt Darstellungsentscheide basierend auf Erkenntnissen verwandter Disziplinen, wie den kognitiven Wahrnehmungstheorien, der Informationsarchitektur und der Computergrafik.

Die meisten Themen der realisierten System-Implementation sind von genereller Gültigkeit für Anwendungen im Bereich Informationsvisualisierung, die kontinuierliche, grosse Datenflüsse von zeitabhängigen Datensätzen in Echtzeit abfragen, kommunizieren, verarbeiten, aktualisieren und simulieren. Vier verschiedene Applikationen verwenden zeitabhängige Datensätze aus den Bereichen Finanzen, Börse und Wissensmanagement und illustrieren die Vielseitigkeit dieser Visualisierungsmethode. Eine ausführliche Analyse bringt die grammatikalischen Prinzipien und Syntax der entstehenden visuellen Muster zum Vorschein und untersucht die Qualitäten und Grenzen der Methode.

Die Forschung wurde im Rahmen des blue-c Projekts an der ETH-Zürich umgesetzt, das eine neuartige Umgebung für virtuelle Realitäten mit tele-immersiven Möglichkeiten realisierte.

Acknowledgements

About six years ago, I became intrigued by the work accomplished by the CAAD group at ETH-Zurich of Prof. Gerhard Schmitt, who researched the far future of computer-aided design and the potential of virtual space with a focus that passed well beyond traditional real-world simulation. From that moment on, I realized how I could merge my interests in technology, creativity, and representation in a meaningful way. I am therefore very grateful to Maia Engeli, who invited me to be a part of that unique group, demonstrated the possibilities of technology on education, and motivated me to fulfill this thesis. In addition, I cannot thank Prof. Gerhard Schmitt enough for giving me the wealth of freedom and continuous enthusiasm, together with many valuable comments, to accomplish this research.

Thanks are also due to the whole interdisciplinary blue-c group, who showed me how motivated people from diverse backgrounds are capable to develop state-of-art technology by competence and persistence. I especially would like to thank Prof. Markus Gross for his constructive support and encouragements in difficult moments.

I would also like to thank the whole Arup Research & Development group in London, which supported me tremendously during my stay and facilitated the most significant breakthroughs of this research. Special thanks to Alvise Simondetti and Matthew Pilgrim for providing me with the chance to have this unique experience, and Dr. Chris Luebkeman for provoking thorough investigation as well as new ideas.

Many discussions and collaborations with friends have helped me considerably to fulfill this work. Special thanks goes to Stephan Würmlin and Edouard Lamboray for patiently solving diverse technical issues, Silke Lang for her everlasting enthusiasm, Christoph Ganser for the spontaneous brainstorming sessions, Remo Burkhard for the German translations and valuable comments, Kuk Hwan Mieusset for her exchange of knowledge and continuous determination, Malgorzata Bugajska for the creative discussions about information visualization and beyond, Christian Spagno for the many constructive conversations and assistance, and Martin Naef for his technical as well as personal support. Ann Heylighen, probably the first person to believe in my research ideas by encouraging me to head for ETH-Zurich, was invaluable yet again by putting an immense amount of time and effort to proofread and comment this work.

This thesis would not have been possible without the love, motivation, and tolerance of my family. What all started many years ago with the thoughts provoked by a Tandy TRS-80 computer in a small Belgian village has developed in a life story taking place in cities such as Zurich and Sydney. Even many hundreds of kilometers away, both my parents and my brother's family (Dave, Kathy, Manou and Nina) have given me the immense freedom and support to pursue my dreams wherever they mentally, but also physically, seem to take me.

Benedicte, without your love, tremendous patience and continuous encouragements this work truly would have stayed in a state of wishful dreaming. Your faith has taken us both to many sides of the world with joy and happiness. Thank you so much for being at my side during all the swirls of life.

Contents

Figures

Tables

Nothing endures but change.

Heraclitus (540 BCE - 480 BCE)

1. Introduction

There is a need for novel information visualization metaphors that are capable to effectively represent dynamic tendencies within time-varying data. Accordingly, one of the main themes underlying this thesis is to demonstrate and prove the potential of data-driven particles for time-varying information display. However, it is still largely unknown what makes such data mapping methodologies effective. In fact, the best metaphors combine creativity and experience, algorithms and aesthetics, and are based upon findings taken from cognitive psychology. Through a process of continuous evolution and reevaluation, successful methods become gradually refined, and stand the test of time.

The infoticle metaphor development combines insights from the fields of architecture, computer science and cognitive psychology. This chapter lists the related motivations and explains briefly the related research contributions, assuming the reader to be familiar with the theoretical and technological aspects of the areas that have greatly influenced this research. These fields, including information architecture, information visualization and virtual reality are described as background literature in Chapter 2.

This thesis has been accomplished within the framework of the blue-c project, which has developed a set of novel tele-immersive virtual reality portals and is described in more detail in Appendix A. As with most research projects that take place on the front edge of technological innovation, novel system capabilities ultimately emerge at the very end of the development process. Consequently, this research focused primarily on exploiting the unique characteristics of immersive virtual reality installations, and the development of an information visualization metaphor that exploits these.

In fact, because of the relative independence of collaboration and tele-immersion, the resulting contributions of this research transcend the context of the blue-c project. Instead, the findings expressed in this thesis are meant to be generally valid for information visualization approaches that represent abstract, time-varying datasets, and eventually might be employed in immersive virtual reality environments.

1.1. Motivation

Intuitive exploration of large, time-varying datasets requires novel visualization techniques that are based upon the scientific findings of cognitive perception. Virtual reality technology holds the promise of exploiting unique human-computer interaction capabilities, even for the field of information architecture. Therefore, effective visualization and interaction metaphors have to be designed that take advantage of the intrinsic qualities of virtual reality technology, and use architectural knowledge to structure and present data as meaningful information. As a potential metaphorical approach, the emergent behavior of particles conveys aesthetic visual patterns that appeal cognitive understanding and human intuition. At the same time, virtual reality software designers should be aware of current user expectations, so that the resulting applications demonstrate similar qualities of interactivity and user engagement as most contemporary multimedia tools.

1.1.1. Information Visualization

In the information society of today, many corporations and scientific fields are continuously accumulating tremendous amounts of data. Recent research has shown that print, film, magnetic and optical storage media produced about 5 *exabytes* of new information in 2002, of which 92% was stored mostly electronically on hard disks (Lyman and Hal, 2003). Well aware of their potential value, many data owners are eager to detect the unknown patterns, tendencies and anomalies that are suspected to be hidden within these datasets. However, although current database technology has made it possible to store and manage huge amounts of real world data in a comprehensive manner, data exploration is still bound to relatively rigid interfacing methods, such as *table-based* or *schematic* queries.

Consequently, the scientific visualization and computer graphics communities have been challenged to develop new *visual metaphors* for understanding, navigating and interactively analyzing abstract information spaces using the latest display and rendering technologies. Successful visualizations are needed that enable users to understand informational relationships through observing spatial phenomena and through exploring the representation space by suitable navigation techniques. For instance, *exploratory data visualization* offers users first insights through a recursive iteration process of tentatively browsing, recognizing patterns and testing assumptions within previously unknown datasets.

Although many visualization techniques exist today, relatively little effort has been spent to create new or adapt existing applications that utilize the unique qualities of virtual reality technology. In fact, most currently employed visualization techniques are focusing on static representations of rigid datasets, overlooking the potential of *motion* as a novel powerful visual cue, and neglecting the importance of dynamic patterns within large, time-based datasets. Furthermore, most visualization approaches mainly focus on the evaluation and refinement of data representation algorithms, hereby underestimating the development of interaction metaphors that are specially adapted to the employed visualization technique itself.

1.1.1.1. Dynamic Data

Sources of abstract data range from simple text analyses, product databases or customer surveys to massive amounts of entries accumulated by DNA analyses or subatomic collision experiments. Moreover, data can have different characteristics that require fundamentally different visualization methods. Classic information visualization methods categorize datasets into: *hierarchical data*, which can be divided in tree-like subcategories, *unstructured data* or *nominal data*, which shows no relationships between their entities, does not possess units and can be described as simply equal or different, *quantitative data*, which is defined by scalar values and can be changed with normal arithmetic operations and *ordered or ordinal data*, which obeys for instance smaller-than relationships.

Dynamic data, also denoted as *time-varying, time-dependent, time-variant, time-based* or *temporal data* can be imagined as a constant stream of different data values. The quantity of collected data either continuously changes by accumulation or deletion, e.g. live news feeds, or consists of a constant number of data entities of which only the data values change, e.g. stock market quotes. A *data update* is classified according to its specific time-dependent behavior, e.g. *continuous* or *discontinuous*, and by its *frequency of change*. Constructing effective representations of dynamic datasets is not trivial, and differs from normal information visualization methods that mainly focus on motionless representations of static datasets. Most traditional time-varying visualization approaches focus on the representation of exact data values, although one should note that the *typology of change* itself reveals valuable data patterns as well. Especially the use of time-dependent *behavior* in visualization applications is still relatively unexplored.

1.1.2. Information Architecture

Currently, the fields of information visualization and human-computer interaction are mainly dominated by the field of *computer science*, although both these areas continuously draw inspiration and findings from several other scientific areas. Many architects, by nature of their multidisciplinary education, have the ability to merge conceptually different directions into universal ideas that are attractive, meaningful and useful. In addition to this combinatory talent, architects typically focus on considerations that are too complex to grasp as a whole with scientific or quantitative means. Aesthetics, space perception and cultural influences are only a few of the guiding motivations that are successfully incorporated into computer applications that have been evaluated by such design experts.

Given the rising importance of information technology in everyday society, the idea of *information processing aesthetics* might answer a need that many users are currently confronting. The typical work situation increasingly transforms into a computer-mediated and pervasive environment that is dominated by productivity-focused applications and communication devices. Hence the growing need for architectural interventions that incorporate space-organizing considerations and inspirations targeted to an enjoyable user experience.

The development of virtual spaces for immersive virtual reality environments has the potential to become the ideal design space for computer-literate architects. Although the traditional architectural boundaries, including construction, gravity and lighting can be easily neglected in virtual space, other limitations determine electronic designs, such as computing performance, intuitive navigation and interaction, spatial awareness and the development of visual metaphors that are effectively comprehendible. Since the cyberspace conception, many researchers have been motivated to invent novel interaction and data visualization metaphors that offer mediated, human-scale and three-dimensional environments generated out of abstract data, in which users can browse and manipulate information intuitively. Similarly, the work presented here is geared to develop a new kind of abstract visual language that focuses on the background of human consciousness and appeals to cognitive and perceptual senses. Ultimately, it might form a meaningful virtual object that inhabits the immersive cyberspace world of the future.

1.1.3. Virtual Reality Characteristics

By utilizing virtual reality technology for visualization applications, one can take advantage of a wider range of human perceptual senses and the ability to virtually navigate through the visual representation. Virtual reality technology has the potential to enhance information visualization applications, as it allows for real-time interaction, stereoscopic perception and the visual analysis of relationships between objects from an unlimited number of perspectives. Consequently, application designers can improve the effectiveness of visualizations by exploiting these powerful capabilities of virtual reality for spatial cognition purposes.

1.1.3.1. Computer Graphics Techniques

Computer graphics is the research field that explores and develops new approaches to modeling, rendering, simulation and scientific visualization by exploiting the latest high-performance computing technology. Most of the resulting research findings are created to mimic physical reality or to overcome typical engineering issues, such as network bandwidth, calculation performance and the like. Some of the simulation methods currently being investigated include the following approaches.

- **Real-time Rendering.** Techniques that enable the creation of images from data files with a high framerate, facilitating real-time navigation and interaction.
- **Particles.** Mathematical points in space that are influenced by internal relationships or external forces, and are animated in time.
- **Morphing & Warping.** Methods for modeling and animations that smoothly transform the shape of one of two given objects into the shape of the other.
- **Inverse Kinematics.** Algorithms that control the motion of an articulated model.

Although most of these developments are not commonly employed beyond the typical computer graphics world of modeling and animation, they could be useable for other, maybe conceptually unrelated, purposes. For instance, these techniques have the potential to offer information visualization designers novel ways to translate their ideas creatively into tools that are enriched by unexplored representation features and efficient perception qualities.

1.1.3.2. Application

Lower-cost computer systems and projectors allow for the production of affordable and user-friendly virtual reality systems to be used in an increasing amount of public venues. As virtual reality technology matures and the total cost of hardware drops, a growing number of people will have the chance to experience virtual reality's unique characteristics of immersion, stereoscopic vision and spatial orientation. Because virtual reality is still seen as *futuristic*, audiences are easier drawn. Many developments within the virtual reality community are likely to enhance the general understanding, meanwhile rendering successful interface elements more common and robust by a process of evolutionary survival.

Virtual reality will move from the specialized, custom-built applications towards mass distribution. Instead of high-end, scientific applications primarily geared towards the specialized few, novel applications will have to be developed for a larger, non-expert user base. These programs will need to focus on specific tasks that are useful for a wide audience, while offering the same quality of user experience and user engagement as one currently encounters in various multimedia applications and entertainment systems.

Simultaneously, this relatively novel display and interaction technology provides content designers with a fundamentally different and more elaborate medium to work with. In theory, virtual reality holds the promise of engaging users with appealing applications, although often it is not the appropriate medium to work with. In effect, application designers should be aware of the critical characteristics that distinguish virtual reality from other new media technologies, also called *virtual reality's five I's* (Sherman and Judkins, 1992).

- **Immersive.** Virtual reality applications should deeply involve or absorb the user.
- **Interactive.** Virtual reality applications should employ necessary techniques that allow to act reciprocally via the computer interface.
- **Intensive.** The user should both be concentrating on and respond to vital information received from multiple sources.
- **Illustrative.** Virtual reality should offer information in a clear, descriptive and illuminating way.
- **Intuitive.** Virtual reality should be easily perceived and the use of virtual tools should be straightforwardly and understandable.

Many virtual reality application designers consider immersive hardware systems as relatively large observation desks enabling the careful and life-like investigation of certain objects that are too complex to visualize by other means. Artists, however, regard such an environment as an enormous blank canvas. In their view, virtual reality offers the unique opportunity to develop designs that demonstrate novel visual languages and definitions, facilitates the exploration of alternative views of reality that are otherwise impossible to perceive, and enables the creation of experiences that are otherwise fully unimaginable.

1.1.3.3. Three-Dimensional

The use of three-dimensionality in applications is still a controversial topic in many scientific areas. The common reasons not to include three dimensions in visualization applications are the lack of computing power for rendering and user interaction, the multifaceted interpretation of a three-dimensional representation in comparison with two-dimensional diagrams, and the complex issues involved with interaction in three-dimensional virtual space. To date, only a few studies exist that compare two-dimensional and three-dimensional approaches for textual information retrieval (Sebrechts, et al., 1999). In theory, the extra physical dimension frees up an extra visual cue that can be spatially encoded, although this code assignment cannot be accomplished arbitrarily, as unrelated patterns might appear due to dimensional redundancy or unexpected perceptual grouping.

Using three dimensions for visualization adds an element of familiarity and reality to the cognitive perception of the representation. Often, three-dimensional virtual environments are considered to offer users larger work spaces, as the corresponding interfaces enable displaying more information without incurring additional cognitive load, because of pre-attentive processing of perspective views and the avoidance of occlusions. Nevertheless, extra navigation tools are necessary to make the space denser and usable (Robertson, et al., 1993), and some rules-of-thumb for three-dimensional interface design are required (Shneiderman, 2003).

Three-dimensional virtual environments typically try to engage the human spatial cognition capabilities such as spatial memory by a process of *mental mapping*, although to date there is no proof that spatial memory works in virtual environments as it does in the real world (Robertson, et al., 1998). Norman (1993) suggests that natural, spatial mappings between items and their spatial location is successful only when (1) there is a natural, spatial mapping between the items and the spatial location, (2) desired items can be located in a minimum of attempts, (3) the number of different items at any single location is small enough to be readily found, and (4) the amount of effort required to try a location, scan its contents, and then try another location is small. Especially the first issue, the mapping of non-spatial aspects of information onto three-dimensional coordinates, is both problematic and potentially powerful. The problem is one of attaching meaning to spatial dimensions in a sensible way so that any user, confronted with *information objects* arranged in space, is able to explore these objects purposively, and interpret their shapes and locations naturally in terms of the concepts one is interested in.

1.1.3.4. Stereoscopic Vision

Fundamental differences exist between three-dimensional environments displayed on the desktop and those that can be perceived in immersive virtual reality installations. Similar differences exist between three-dimensional and stereoscopic perception or, translated into contemporary technological means, three-dimensional representations on flat displays (also called two-and-a-half-dimensional) and on virtual reality installations (Gross, 1994). While both offer the perception of depth, only virtual reality allows for the user to immerse oneself into the representation. In addition, stereoscopic vision supports the estimation of relative objects sizes and the distances between objects. In contrast, one has to move or rotate the view on traditional displays to understand the diagram by observing the occlusions and relative speeds of the objects in the scene, a phenomenon also known as the *depth cue-of-motion parallax*. The concept of stereoscopic vision, as experienced in immersive virtual reality environments, enhances all these aspects, as it offers the capability to cognitively detect motion in all dimensions, even depth. Consequently, this thesis proposes immersive systems as ideal environments to demonstrate an information visualization technique that is primarily based upon the concept of three-dimensional space, motion, form recognition and direct interaction.

1.1.3.5. Physical Immersion

The human appeal of *immersion* originates from the illusion of a three-dimensional reality and from the capability for direct interaction within that reality. The concept of immersion promises users the theoretical ability for a perfect sense of spatial orientation and three-dimensional form recognition. These specific characteristics create the opportunity for the *spatial location* of data as an organizational principle. The oldest example of managing data in a spatial way is known as *Method of Loci* or *Method of Place* and originated from Simonides, a poet and teacher of rhetoric who lived in ancient Greece about 500 B.C. This classical Greek method of virtual experience relied on the internal mental construction of virtual spaces. It tries to mnemonically assign abstract concepts to sequential, static positions in space for better memorization. Virtual reality technology facilitates a similar concept, provoking the spatial memory of users as a framework for efficient memorization and orientation.

In an immersive environment, one can directly perceive and manipulate three-dimensional objects inside the scene instead of interacting with abstract interface elements. Because one can change the viewpoint by physical action, for instance by head movements, space perception and orientation are enhanced. Especially the use of *peripheral vision* offers users a good comprise between global overview and detailed analysis. Nevertheless, most large-scale virtual reality installations still require a certain main direction towards the front screen, so that orientation towards the ever-changing coordinate axes in virtual space might be rendered difficult. Therefore, empirical evidence has shown that virtual reality is especially suited for visualizations that require only a relatively small physical frame of attention, offering users a continuous contextual overview.

1.1.4. Particle Metaphor

In all visualizations, graphical elements are used as a visual syntax to represent meaning, and the combination of these encoded representations results in a *visual metaphor*. Notably, the most influential decision in developing new information visualization methods is choosing a suitable metaphor to translate data values into specific visual counterparts.

1.1.4.1. Abstraction

As the field of information visualization becomes more mature, one can observe an increase in conceptual abstraction of mapping mechanisms. This movement contrasts with those visualizations that prefer to use consistent, real-world or so-called *natural metaphors* exploiting the familiarity of direct analogy with the real physical world. One of the most commonly used metaphors uses the concept of cities to visualize abstract information, such as computing networks (Abel, et al., 2000) or software source code (Knight and Munro, 2000). These approaches are motivated by the hypothesis that, based upon prior knowledge of how metaphorical elements are spatially arranged in the real world, a user can directly relate to the way the information is presented in the virtual environment (Dieberger and Frank, 1998).

The ultimate visualization goal consists in shifting the cognitive processing needed for navigation and visual interpretation to the subconscious level, exploiting the knowledge and experience employed in everyday actions. However, older media technologies such as television and Internet have clearly shown that real world analogy metaphors are primarily used around the time of global introduction, mainly to present people the novel capabilities in a simple and understandable manner. Once people get accustomed to the features of these media, certain elements, mostly explored by the avant-garde art crowd, become popular and merge into the daily reality experience. For instance, flash-back story-telling in movies evolved from text legends or blurry picture morphs into sudden, unexplained cuts within the story, and engaging websites already seized to present users with analogies of doors, windows or real-world city lay-outs. Generally speaking, it is easy to understand a visualization that is faithful to everyday objects, although it is a challenging task to invent a new visual structure that is unique to information space characteristics (Wakita and Matsumoto, 2003).

In contrast, *creative structure metaphors* visualize things and phenomena in a way totally different from their original state. For instance, the fields of Informative Art and Ambient Displays (see Section 2.1.3. Background - Information Architecture - Info-Aesthetics) show how the introduction of digitized media facilitated the translation of information into a more subtle sensory language that appeals to the unconscious senses of human intuition. As a truly ultimate and purely fictional example, the movie 'The Matrix' (Wachowski and Wachowski, 1999) shows in a fascinated way how tremendous amounts of data, here to simulate and sustain a dual virtual reality for millions of users, can be abstracted into a single, dynamic interface resembling a dynamic collection of dropping green signs, which is apparently comprehensible by an expert observer.

Figure 1.1 and Figure 1.2 show two examples of applications that use creative metaphors to provoke the perception of hidden structures inside data. 3.D.H.T.M.L. is a three-dimensional visualization of web page structures that runs in parallel to the normal web browser on the computer desktop (Vande Moere, 2001). It extends the idea of translating one-dimensional HTML-coded text and its two-dimensional browser layout rendering into the third dimension, by translating the web page's structural qualities and content characteristics into a visual, architectural language. Consequently, by looking at the shape of the virtual, three-dimensional artifact, one is able to recognize the functionality of the web page. The system also enables users to truly model an aesthetic three-dimensional shape interactively and analyze the qualities of the resulting web page counterpart in real time.

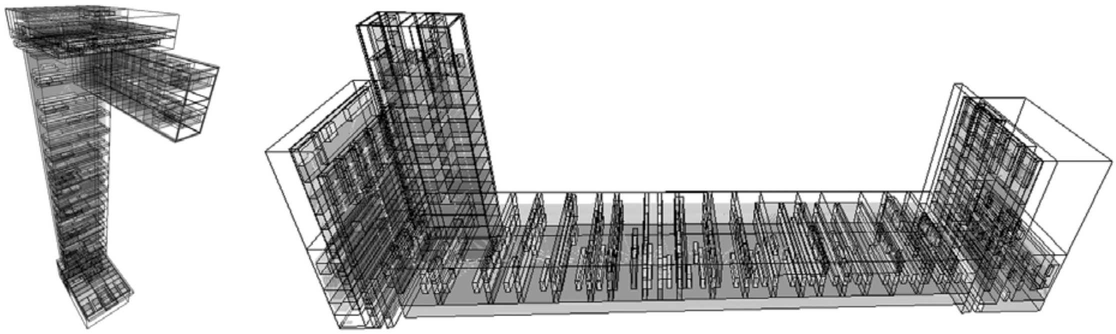


Figure 1.1. 3.D.H.T.M.L.

Similarly, the Virtual Library maps content category explorations of library visitors onto the direction of lines, called *Paths of Knowledge* (Bugajska and Vande Moere, 2001). Users are able to explore and click the three-dimensional lines that cross their own created path tunnel, and become intuitively inspired by the previous searches of other users with similar interests.

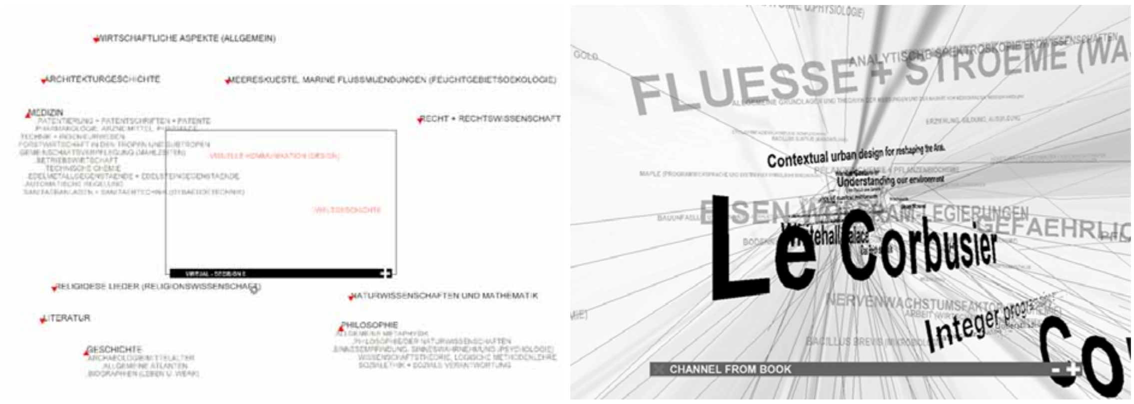


Figure 1.2. Virtual library.

1.1.4.2. Metaphor

Some researchers suggest that the potential power of virtual reality originates from the strength of its metaphor, and from the fact that the interaction is much more natural than with many other computing systems (Pettifer and West, 1997). At least in theory, three-dimensional metaphors have a much broader scope, as they engage the human perceptual and spatial capabilities used in everyday life for comprehending the surrounding environment. It is the very potential to directly engage these cognitive capabilities that constitutes the defining characteristic of virtual reality. Therefore, the role and effective management of suitable metaphors are of key significance and have to be thoroughly investigated.

Particle systems are a relatively new computer graphics technique which only recently has become cheap in calculation cost. Particles have the ability to behave in complex ways, and can be perceived in stereoscopic environments in large amounts without the problem of conjoin or overlapping. Many particle animations are visually seductive because they continuously behave in an unpredictable and ever-changing manner. Nevertheless, human cognition relates the character of dynamic trajectories to meaningful emotions (see also Section 3.2. Infoticle - Cognition), or intuitively maps the concept of dynamic data streams onto the flow of particles. Accordingly, particles are used in many metaphorical descriptions of complex phenomena, such as planetary systems, space galaxies, electron dynamics and point explosions. Furthermore, particles are theoretically able to represent very large amounts of time-based data because their updates can be *non-exact*. By focusing on differences in motion typology instead of exact absolute positions, particles have the inherent capability to be updated non-continuously while still representing a smooth motion.

1.1.4.3. Emergent Behavior

Particles combined with local rules can result into *emergent behavior*, meaning that they act in ways that transcend what they have been explicitly programmed to do. This fact inherently implies that a behavior designer cannot reasonably predict the exact visual outcome of the controlling algorithms. This phenomenon holds great potential in the context of exploratory information visualization, where the user is unaware of the data structure, and often does not have a predefined idea of what specific data patterns should be analyzed. Consequently, emergent behavior generation for information visualization acts in a self-organizing way, displaying meaningful data patterns both at a micro-scale, by local interactions between atomic entities, and at a macro-scale, by global emergent behavior patterns.

1.1.4.4. Conclusion

The combination of the above mentioned considerations regarding visualization abstraction, visual metaphor seductiveness and the unpredictability of emergent behavior generation was the primary motivation to choose the concept of *particle motion* for visualizing abstract data. Consequently, various similarities arose between the particle usage in information and scientific visualizations during the development process. Further research also showed that particles can be employed as a stable framework to develop three-dimensional shapes by specific computer graphics algorithms such as *implicit rendering* and the *marching cubes algorithm*, which potentially could further enhance the visual insight in the data representation.

1.1.5. Motion for Information Display

Current technology has made it possible to implement the concept of *motion* as a true alternative to static representations. In all forms of digital media, animation is composed of sequences of static images altering rapidly enough to create the illusion of an ever-changing continuum. Motion is conveyed through a process called *animation*. It is a dynamic visual statement, in which form and structure evolve through movement over time.

1.1.5.1. Information Display

According to Lynn (1999), the recent introduction of advanced motion simulation tools will finally allow for the exploration of forms that are generated by algorithms and interacting force vectors. Instead of a frame through which time and space pass, architecture can be modeled as a true participant determined by movements and forces. The resulting shapes can be judged according to both formal and functional criteria. Notably, all these factors hold great potential for information display.

Most current information visualization applications rely solely on *static* graphical representations, using generally accepted visual cues such as shape, symbol, size, color and position to encode abstract information. Motion however, is currently applied in an unplanned manner, often to display a certain transition between two distinct, static states. Already more than a decade ago, some authors have foreseen applications that use animation along other media to make computer interfaces more enjoyable and comprehensible (Baecker and Small, 1991). Some researchers even consider the use of animated graphics to facilitate learning (Morrison, et al., 2000). Although animation seems to be a perceptually rich and efficient display technique, little is still known about its attributes for information display. Bartram (2001) describes the use of motion to convey information in user interfaces as particularly interesting for several reasons.

- **Perception.** Motion is perceptually efficient as the human perceptual system is pre-attentively sensitive to motion.
- **Interpretation.** Motion has a rich interpretative scope since simple actions can be cognitively interpreted as highly sophisticated behavior.
- **Bandwidth.** Motion has more potential coding bandwidth because it is still under-used as mostly it is the object moving, not the nature of the movement, which carries the meaning.
- **Technology.** Thanks to the advances in graphics technology, it is increasingly easy to compute motion, and it takes little extra screen space to display.

Notably, the use of motion to distinguish patterns in abstract data is at present only a small research topic in the field of information visualization.

1.1.5.2. Pre-Attention

Motion is a *pre-attentive* feature, as it is processed automatically by the human perceptual system without any conscious focus of attention. Pre-attention denotes *preconscious* as opposed to deliberate processing of information. In general, anything that is processed at a faster rate than 250msec per item is considered to be pre-attentive, because within this timeframe an eye movement to a new spatial location cannot be made (Ware, 2000). The use of pre-attentive processing in visualization metaphors is desirable, as it shows things *at a glance*, and consequently frees up cognitive resources for other tasks, such as navigation and interaction.

The combination of visualization methods with scientific insights from *cognitive psychology* could lead to visualization systems that support rapid, accurate and effective exploration and analysis of large, complex, multi-dimensional datasets. Cognitive facilitation allows for much of the analysis work to be performed effortlessly, without requiring focused attention. Research in this direction suggests that pre-attentive and other visual features such as hue, orientation, intensity, size, curvature and line length can be merged to create visualizations that allow for high-speed multi-dimensional data analysis in real-time (Healey, et al., 1995). However, it also shows that care must be taken in assigning data elements to pre-attentive features to avoid the creation of visual interference effects (Healey, 1999).

1.1.6. User Engagement

Over the past few years, developments in information visualization have increasingly focused on *task metrics* measurements, *usability* enhancements and the generation of organization algorithms for massive amounts of raw data. As research increasingly dealt with such efficiency and convenience considerations, the specification of a corresponding interaction design was neglected and even became separated during the development process. At the same time, many new media interfaces have been implemented that seem pleasing to the eye and joyful to use, and clearly considered the concept of user experience during the design stage.

Application designers agree that users not only perform a certain task with a fixed goal, but also like to experience a pleasurable joy alongside it. For instance, while surfing the Internet for bits of valuable information, the whole process of exploring and browsing, with its anticipatory emotions of expectation and surprise, and the potential to be inspired by unexpected related subjects, becomes at least as important as finding the result itself. Issues of affective and emotional responses to interactive systems are exceedingly important within the communities of human factors, product design and design research. The need to understand and create emotional and aesthetic connections between people and products seems to be increasing (Westerlund, 2002). Like a garden design with vantage points that hide, reveal and accentuate a series of features, well designed *information landscapes* should provide a meaningful context, be inviting and comfortable, and their exploration should be a true delight (Small, 1998). Some usability experts stress the importance of beauty, fun and pleasure. They even prove that attractive things work more effectively by the combination of cognitive and affective factors (Norman, 2003). Many contemporary multimedia applications show an increasing tendency towards longer enjoyment times, influenced by concepts such as entertainment, pleasant feelings and coolness.

1.1.6.1. Definition

Laurel (1991) introduced the term *user engagement* and emphasizes the importance of engagement and pleasure in multimedia applications. She presents the possibility of blurring the edges between applications and interfaces by incorporating insights and techniques from other disciplines. Accordingly, art can be considered as a new and rich source of useful knowledge, next to computer science and traditional interface theory. Only recently has the notion of engagement begun to be formalized, so that the concept has rarely been studied in the information systems literature (Chapman, et al., 1999). Even within the field of information visualization, some researchers started to consider whether *effectiveness* and *aesthetics* might be related, and that much can be learned from studying human psychophysics, art theory and art history to create effective and engaging visualizations (Healey and Enns, 2002). Engaging systems are typically measured by attention focus, curiosity and intrinsic interest, and have been described as drawing users into the activity, capturing their interest and attention (Laurel, 1991), and are capable to seduce users (Skelly, et al., 1994). Direct engagement emphasizes emotional as well as cognitive values. It conceives human-computer activity as a designed *experience*, reconfiguring application and interface design as a single integrated process.

1.1.6.2. Mental Immersion

Since the successful introduction of game consoles and multi-user role-playing games in everyday society, the concept of being mentally immersed while interacting in synthetic, computer-generated worlds became socially accepted. *Mental immersion* in virtual reality applications is often desirable and critical, but not a definition factor of this technology, which is primarily based upon *physical immersion*. Following the theories of hypertext narratives, also better known as online storytelling, user engagement should be applied together with a form of mental immersion, to result in a form of *pleasure* (Douglas and Hargadon, 2000). One should note that immersion and engagement are different, as immersion absorbs one mentally within the *flow* of a schema, while engagement is provoked through the recognition of a conflicting schema from an outside, observer perspective. Merged together, they provoke the human capability to guess the intention of the perceived schemas. Immersion typically is satisfying as long as unpredictable events are introduced, while engagement stays pleasurable when schemas are subverted but alternatives are offered within a visible framework. Elements of surprise are introduced with *interactive* features, so that the user experience becomes unpredictable and the final outcome unforeseen. Also, many multimedia applications gain much of their immersive power from spectacular effects, just as spectacle tends to enhance participatory narrative in order to retain attention (Murray, 2000).

Immersion can also denote a certain level of engagement, and as such indicates the successfulness of the communication within a virtual world (Sherman and Craig, 2003). The active design of computer information space in virtual environments comprises the design of *spatial flow*. Ancient civilization developed an art of placement for enhancing and improving the quality of energy flow through space, also known as the art of *feng shui*, which redresses the imbalances of cognitive systems theory (Heim, 2000).

1.1.6.3. Conclusion

Invoking immersion and user engagement in application design results in more satisfied users, who feel more rewarded and become intrinsically motivated to learn and spend more enjoyable time on specific tasks. Therefore, there have been calls to create systems that recognize the achievement of engagement as an important goal in application design. Obviously, a balance between usability and enjoyment, two generally conflicting goals, has to be determined.

1.2. Contributions

This research proposes a novel information visualization metaphor that uses the motion characteristics of particle systems to convey the update typology of large, time-varying datasets, and simultaneously facilitates a high degree of interaction and navigation within the resulting data representation. This visualization technique combines the unique qualities of virtual reality hardware technology such as immersion, stereoscopic vision and cognitive perception with human-computer interaction techniques into a set of effective applications that hold promise of revealing dynamic data patterns in a user-engaging way. Several prototype systems for interactively exploring abstract time-varying datasets in virtual reality environments are presented. They are analyzed as real-world case studies that utilize the features of motion and emergent behavior generation for information visualization purposes. Consequently, the main contributions of this thesis are situated in the areas of information architecture, information visualization and computer graphics.

1.2.1. Visualization Metaphor

Development of a novel information visualization metaphor, based upon insights taken from information architecture, computer graphics and cognitive psychology. This representation method maps time-varying data updates onto the three-dimensional movement and kinetic behavior of data-driven particles, coined infoticles. A visual typology of recognizable patterns emerges that represents data update characteristics and corresponding dynamic data value alterations.

To effectively represent the dynamic characteristics of large, time-varying datasets, new visualization metaphors are needed that to meet the following objectives.

– Allow people to *browse* and *explore* large information spaces in an effective way.

– Let people *freely* and *interactively* filter, cluster and analyze datasets according to informational values.

– Allow people to *model* their individual three-dimensional information exploration environment by combining *spatial reasoning* with information querying.

– Visualize the characteristics of *dynamic, time-varying* dataset *changes* by using cognitively recognizable *motion typologies*.

– Facilitate the manipulation of the speed and time direction of the animated *timeline simulation*.

– Employ relevant research findings of *cognitive science* to effectively exploit the unique qualities of stereoscopic and three-dimensional environments in combination with motion and shape perception.

– Combine one-, two-, three- and four-dimensional virtual objects to intuitively distinguish *interface functionality*.

The infoticle metaphor widens the search for or recognition of relevant data trends, breaks fixations on single data model hypotheses, and considers various possible visual patterns to explain these. It is a real-world working solution that draws a user's attention to valuable and informative relationships in time-varying data, even if the users are not aware or do not know to look for those data patterns explicitly beforehand.

The use of motion as a means of conveying information is implemented through the concept of data-driven, animated particles, coined *infoticles*. They express the update characteristics of changing data values by altering dynamic particle properties, such as color, speed and direction. A set of tools, consisting of *sources*, *forces*, *filters, shapes* and *traces*, supports the user to interact with the graphical representation and to interpret informational values of the dynamic properties. Moreover, these features enable users to *design* their own three-dimensional information environment, and to filter and query data by *modeling* infoticle tools in meaningful, three-dimensional constellations.

Local behavior rules associated with each single infoticle determine the dynamic actions that correspond to each dynamic data value change. These specifications are based upon *external* influences generated by the tools within the virtual scene and upon *internal* data-dependent relationships among the infoticles themselves.

This metaphor allows for the visualization of large quantities of time-related data, as it is capable to cope with different data granularities, timeframe sizes and data inconsistencies. It enables users to explore and analyze data in a non-quantitative way, by showing both global, general trends from a macro-perspective as well as the performance of individual data entities. This analysis takes place without loosing the global contextual overview over both the actual time state and the spatial representation.

In short, this thesis will propose the infoticle method as useful in visualizing dynamic update characteristics of complex information collections because of its use of pre-attentive and interpretative perceptual properties and its ability for a high degree of user interaction by modeling personalized spatial configurations.

1.2.2. Time-Varying Data Update Characteristics

Recognition and evaluation of various meaningful characteristics of time-varying datasets, such as update frequency, update frequency evolution, relative data change, data change history, for individual data objects and in the context of the whole dataset. The description of general insights for visualizing dynamic update characteristics of time-varying datasets using motion properties and self-organizing local rules.

These guidelines demonstrate how data-driven particles are able to represent the so-called *delta* or *change* in time-varying data, instead of the traditional focus on visualizing exact data values. These insights have been discovered and exploited during the infoticle metaphor development process, and are useful for evaluating existing or inventing novel metaphors that visualize *how* the dataset values change and evolve in time, instead of solely visualizing the exact data values themselves.

- Data pattern recognition by *breaking metaphor consistency*.
- Exploitation of *spatial adaptation time* to cluster similarities in *data update history evolutions*.
- Usage of *motion characteristics* and *motion typologies,* instead of objects moving, to visualize dynamic, abstract information.
- Representation of *data update characteristics* and relative *data value changes*, instead of the updated data values themselves.
- Caching and processing of *parallel sequential datasets* in real time.

1.2.3. Real-Time Data Processing

Implementation of an effective mechanism to efficiently query, process, simulate and represent large streams of time-varying data from a remote database in real time. As the infoticle metaphor depends on the continuous retrieval of large quantities of time-ordered, remote data and the quasi-immediate evaluation and dynamic behavior generation thereafter, several software implementation and calculation performance issues were encountered and have been solved.

Research in cognitive science has shown that in order for an animation to be effective, a natural interaction with the timeline simulation and the manipulation of the time direction and speed are required. In addition, all animations need to be smooth and continuous at all times. Several issues are described that facilitate such an interactive and dynamic reevaluation of time-varying data simulations that are never interrupted by any external calculation or information handling processes.

Several possible implementation solutions are described that deal with the handling of large streams of time-varying data from a remote database in real time. In effect, the querying, caching, conversion, ordering and communication of such datasets must not influence the application performance nor the smoothly animated data representation, and requires the implementation of *parallel processing*, *data caching, memory sharing,* and *procedural optimization*. In addition, depending on the internal update frequency, *singular* or *parallel sequential* datasets need conceptually different caching, evaluation and distribution algorithms to represent the reoccurrence of equal data objects within the same update timeframe.

1.2.4. Virtual Reality Application

Development of several information visualization application prototypes for an immersive virtual reality environment, with special attention to the aspects of stereoscopic vision, human-computer interaction and user-engagement.

Several information visualization application prototypes are proposed that employ the data-driven particle metaphor and exploit the unique characteristics of immersive virtual reality environments. The application design takes into account the complex technical requirements of immersive virtual reality technology, instead of converting common three-dimensional desktop application insights. The applications are adapted to virtual reality input devices, and take advantage of the qualities of immersion, stereoscopic perception and spatial awareness.

Additionally, an intuitive user interface has been developed that does not break the three-dimensional illusion or occludes the visualization itself. These prototypes demonstrate how the infoticle visualization metaphor can be employed to offer immersive virtual reality users the aesthetic and engaging qualities that can be found in many contemporary multimedia applications.

Furthermore, information architectural insights are employed to increase user-engagement and to present the information within a virtual space through subtle changes that can be processed in the background of awareness.

1.2.5. Case Study Evaluation & Analysis

Development of the infoticle metaphor in four different conceptual forms, called Modeling World, Galaxy World, Electron World and Boids World. Evaluation and analysis of the infoticle concept by implementing real-world applications in different areas, namely knowledge management, finance and stock market exchange. Recognition of the significant infoticle representation characteristics, including the emergent visual grammar and the corresponding informational meanings. Analysis of the capabilities, limitations and ideal dataset features of the proposed visualization concept.

To prove the versatility of the infoticle metaphor, different working prototypes were implemented in a demo-oriented fashion. These visualization scenarios use the general concept of data-driven particles, but are adapted to specific dataset characteristics. Each of these applications focuses on different aspects of motion simulation, behavior generation and user interaction. Furthermore, an empirical analysis categorizes the emergent features of each infoticle representation, and identifies the unique capabilities and limitations of the visualization method at the current implementation state.

1.3. Organization

This thesis starts with the main motivations for combining specific technological, algorithmic and cognitive findings into a single visualization metaphor.

This is followed in Chapter 2 by a short description of the four main related fields which have inspired this work: information architecture, virtual reality, information visualization and the particle system concept. This chapter also discusses the scientific work that is closely related to the research contributions, including other information visualization applications in the realm of immersive virtual reality.

The infoticle concept is explained in detail in Chapter 3. It first describes the conceptual basis and the research findings from cognitive science that relate to motion perception, stereoscopic vision and behavior generation. Next, it introduces the conceptual structure and the exact behavior rules that build the core of the data-driven particle methodology. It describes in detail the adopted design rationale in the context of related insights from diverse disciplines, ranging from human cognition over information architecture to computer graphics research. Furthermore, the generally valid implementation issues that relate to the development of a time-varying representation method are revealed.

Chapter 4 describes the development of the different prototype applications that were implemented according to the infoticle metaphor, using real-world datasets from the worlds of finance, stock market exchange and knowledge management. This chapter mentions the various infoticle method adaptations that depend on the specific dataset characteristics, and includes the analysis of some empirically discovered usability issues.

In Chapter 5, the infoticle concept is analyzed and its prototype applications are compared, out of which specific grammatical principles generated by the data-driven particles emerge. Furthermore, several issues are evaluated regarding the use, qualities and limitations of the proposed visualization method.

Finally, Chapter 6 concludes with a discussion of the research contributions and the future possibilities of the infoticle concept for time-varying information visualization.

2. Background

The research presented in this thesis combines insights from diverse disciplines, and is enclosed by a conceptual triangular framework, with its three imaginative corners denoting the fields of architecture, information visualization and computer graphics. Merged together, these concepts constitute the general, inspirational background of the infoticle methodology.

This chapter lists the scientific investigations that show significant conceptual similarities with the proposed visualization metaphor, either regarding the information visualization technique, the fields of application, the information architectural reasoning, the technological means used or the particle system approach. In addition, several real-world examples that utilize particles, time-varying datasets or other similar methods are discussed.

2.1. Information Architecture

Since the first use of the term *architecture of information* in 1976 by Richard Saul Wurman at the national conference of the American Institute of Architects (AIA), many conceptual connections between architecture and information handling have emerged (Vande Moere, 1998).

2.1.1. Cyberspace

One of the earliest proposals for *information space browsing*, and maybe the oldest academic predecessor of the *cyberspace* concept, describes a *spatial database* that exploits the power of human spatial memory to locate information (Fields and Negroponte, 1977). Driven by the first description of cyberspace (Gibson, 1984), many scientists, science-fiction writers and media artists have envisioned a virtual world filled with social life and interaction. Several fictional descriptions, for instance Hollywood movies such as The Matrix (Wachowski and Wachowski, 1999), Johnny Mnemonic (Longo, 1995), Disclosure (Levinson, 1994) and Tron (Lisberger, 1982), articulate this environment as a metropolis of pure data constructs, able to stimulate the human senses in many spectacular ways. Science-fiction authors typically illustrate cyberspace as a live, three-dimensional, shared environment. Often, cyberspace is dynamically created by specific electronic protocols and autonomous *agents* that encode and exchange information in real time, enabling users to locate, retrieve, manipulate and communicate information. Gibson's innovation lies in the description of how a user might not only process information but visualize, feel and interact with it (Merritt, 2000).

Due to its success in various disciplines, the cyberspace concept became linked to many other imaginative synonyms such as the *metaverse*, *dataspace*, *matrix*, *electropolis*, *cyburbia*, *cybertecture*, *hypermedia* and the Internet, rendering the actual term technically unimportant. During the conceptual rise of the cyberspace phenomenon, many researchers were convinced that the field of architecture, with its expertise in areas such as space perception, building history and functional organization, was the ideal professional and academic field to design these virtual environments according to cultural and public needs. So-called *cyberspace architects*, schooled in both art and computer sciences, were destined to be the ideal experts to determine and program the needed simulation software algorithms that translate abstract data into suitable and interpretable forms (Benedikt, 1991, Mitchell, 1995, Negroponte, 1996). In this context, cyberspace can be interpreted as an immersive environment consisting of experiential data structures, offering users a model for interactive data and information-organizing aesthetics.

2.1.2. Architecture

In a December 1998 interview, Richard Saul Wurman explained the term *information architecture* as the combination of three fields: technology, graphic design and writing or journalism (Davis, 1998). Since then, the field has matured and the term has been refined beyond Wurman's explanation, yet the exact definition is still subject of debate (Dillon, 2002). Currently, information architecture also denotes an established professional field that deals with the functional, technical and content specifications of multimedia applications such as websites and interactive information interfaces. It conceptually differs from *information design* as it concentrates on structural rather than presentation issues, and is stripped of the architectural meaning in the classical sense, which points to the skillful designer of social spaces who received a background in theory, history and functionality of issues such as presentation, interaction and form interpretation.

However, many computer-literate architects still consider the term information architecture to denote the usage of *architectural principles* to design and structure data in such a way that the retrieval and exploration of information becomes more effective (Engeli, 2001). Traditional architects have to be competent in both structuring buildings and preparing effective presentations of space, so that people are able to experience the carefully designed qualities of well-considered architectural compositions. Information architecture is not that different in that respect: it addresses the structural organization and effective presentation of data into valuable and meaningful information. This implies that information is considered to be raw material and that the only reality is a virtual one, turning the computer into an instrument, infrastructure and design environment (Schmitt, 1999).

Notably, both architecture and application design have the wish "to create livable, workable, attractive environments" (Mountford, 1992). Both fields consider all functional requirements, often referred to as *constraints*, which a design has to fulfill. These are collected into a so-called *program*, a term that started to play an important role in the architectural movement towards functionalism, following Louis Henry Sullivan's concept of *form follows function* and Le Corbusier's comparison of a house with a *machine for living*. Just like architects have to be aware of the occupant's needs, organize those needs into a coherent pattern, and design a space that lives up to these expectations, information architects gather, organize and present information to serve a well-defined purpose. Accordingly, using the semiological features of architecture and informatics, one can consider real-world information structures as architectural plans and urban forms (Chalmers, 1999). Next to physical reality and time, information can be declared the *5th dimension* of architecture, consisting of architectural knowledge, building regulations, design information and sensory data (Schmitt, 1999). Architecture is then considered as a synonym of structure, and information architecture as the field that organizes and presents the ordered, accumulated data. Consequently, one can imagine the concept *form follows data*, or the use of bits as virtual bricks, not only in the context of exploring formalistic data-mapping principles, but also to detect and aesthetically translate inherent structures hidden inside abstract datasets, and to learn the qualities of new visual languages that represent these.

When considering architecture as a theoretical field that deals with the design of social spaces in which humans work, meet and play, the possible field of application can be broadened from atoms to encompass bits as well. Such digital spaces still enjoy an increasing use in everyday life and have a field of application that reaches from simple text-messaging chat rooms between mobile phones over corporate knowledge management tools to three-dimensional virtual representations of real cities. Notably, in the information society of today, most users enter the digital realm either for social activities or to perform some sort of information processing. They have to continuously interact with abstract interfaces that both facilitate their actions and make them aware of relevant contextual information. As these electronic spaces are freed from all physical constraints, the cyberspace architects are confronted with a tremendous range of design possibilities that need a set of structural guidelines (Bugajska, 2003).

The movement towards the integration of information handling in architecture and art challenges the boundary between aesthetic and commercial applications. After almost a decade, it has led to the establishment of several graduate courses, such as *virtual architecture*, *algorithmic architecture* and *evolutionary design,* currently being taught at architectural departments of various universities.

2.1.3. Info-Aesthetics

According to Manovich (2001b), *new media* represents the convergence of two separate historical trajectories: *computing* (e.g. calculation and communication of data) and *media technologies* (e.g. electronic generation and storage of images). This synthesis has lead to the translation of all existing media into numerical data accessible through computers, resulting in graphics, moving images, sounds, shapes, spaces and texts that have become computable. *New media art* in turn uses the computer as a medium, generating art objects that are inherently programmable. In this field, the connection between content and form, or content and interface, becomes always motivated and designed, merges into one entity and no longer can be taken apart. Consequently, this combination calls for an appropriate translation of content into interpretable forms, a common need that is also recognized by the fields of both information architecture and information visualization. However, instead of mainly focusing on the structural representation of datasets or the usability of data visualizations, new media artists rather deal with the aesthetics and emotional qualities that can be transferred to potential users.

These insights have led to the conception of *info-aesthetics*, which tries to theoretically analyze the aesthetics of information access as well as the creation of new media objects that *aestheticize* information processing (Manovich, 2001a). Info-aesthetics even considers currently existing computer interfaces, often not designed by artists or interaction experts, as true forms of art, and calls them the modernistic, monumental artifacts of contemporary information society that are mostly designed as reconfigured forms of older cultural visual languages.

As a further example of art influences in the field of information visualization, the movement of *informative art* consists of specially designed, computer-augmented, or -amplified works of art that are not only aesthetical objects but also information displays, in that they dynamically reflect information about their environment (Redström, et al., 2000, Holmquist and Skog, 2003). The employed visual metaphors typically include the manipulation of known art styles or works of art, mapping parts or properties of the composition onto different sources of data, and letting them change over time. In a similar way, the concept of *ambient displays* ports information visualization to the built environment. Ambient displays are abstract and aesthetic side-lined displays portraying non-critical information in the periphery of a user's attention. Following the movement of morphing architectural spaces and surfaces into interactive 'interfaces' (Ishii and Ullmer, 1997), such displays are capable to present information within a space through subtle changes in light, sound and movement, which can be processed in the background of awareness (Wisneski, et al., 1998).

2.1.4. Related Work

Visualizing abstract information using architectural principles is still being explored by various researchers. The following section demonstrates real-world examples that have been accomplished so far, and consequently have indirectly functioned as inspirational sources for the infoticle visualization metaphor development.

2.1.4.1. Architecture

Many contemporary projects illustrate the shift of architecture towards the realm of electronic spaces. One of the most famous examples of virtual data-processing architecture is the virtual trading floor of the New York Stock Exchange (also called 3DTF) (Scanlon, 1999, Moltenbrey, 1999), which visualizes trading activity and real-time data streams by three-dimensional line diagrams. It is designed by Asymptote Architecture, which also created the three-dimensional, interactive artifacts representing various statistical datasets of the United States as part of the Understanding USA website (Wurman, 2000), and developed the Virtual Guggenheim Museum (Drutt, et al., 2002).

Other examples of the increasing use of abstract data representations inspired by architectural principles include Datacity (Maas and MvRdV, 1999), representing statistical data from The Netherlands in the context of urban planning; the ETH-World competition of a virtual campus (Carrard, 2001); many designs of AMO, the new branch of OMA that deals with conceptual and strategy thinking instead of pure realization; and the context analysis diagrams generated by UNStudio (e.g. density studies, time usage, flow diagrams, etc.) (van Berkel and Bos, 1999). These examples illustrate that, similarly to traditional information visualization experts, architects continuously require the communication of valuable information to clients, and typically build their designs upon abstract dynamic datasets.

Some commercial firms focus on the architectural design of virtual information spaces, and have implemented various projects: info.scape, an interactive three-dimensional hypertextu(r)al interface presenting the politics of the European Commission in the field of information communication technologies, designed by LAB[au] (2003); electroscape and knowscape, a three-dimensional browser environment representing collective knowledge architecture, by the electronic architecture firm Fabric.ch (Babski and Carion, 2003); diverse real-time data visualizations of Unified Field, including Milkyway, a time-varying information visualization using thousands of moving particles (unified_field, 2003); and some of the projects implemented by Art+Com (2003) and V2.nl (2003).

Several individuals focus on converting information into cyberspatial form, such as Anders (1999), who maps the interdisciplinary creation of dimensional, spatial cyberspaces; Novak (2003), who translates digital inputs into three-dimensional shapes and Engeli (2001), who merges architecture and information processing in various creative ways, mainly for educational purposes.

2.1.4.2. Informative Art

Informative art pieces typically do not provide viewers with exact information that can be instantly understood, but rather supply visual impressions of data changing over time (Holmquist and Skog, 2003). Vice versa, some approaches start from the information visualization side and attempt to merge effectiveness and engagement by drawing inspiration from artistic works (Healey and Enns, 2002). Accordingly, Arc Diagrams are very simple representations of music, text or compiled code and show interesting artistic aesthetic qualities (Wattenberg, 2002). Many other examples can be found yearly in the ACM SIGGRAPH sketches and applications category, at the MIT Tangible Interfaces Group, e.g. information reactive water lamps and pinwheels (Dahley, et al., 1998) and in many contemporary musea around the world, e.g. interactive genomic data visualization (Adams, et al., 2002).

2.2. Information Visualization

Visualization, the representation of data graphically rather than textually, uses the high-bandwidth human perceptual and cognitive capabilities to detect patterns and draw interferences from visual form. Information visualization has emerged over the past fifteen years as a distinct academic discipline inspired by the fields of computer science, psychology, semiotics, graphic design, cartography and art. Many different areas within the scientific field of information visualization can be distinguished today. All attempt to address various research subjects: the use of *visual metaphors* for translating datasets and the *interaction paradigms* that enable effective navigation inside the resulting representations. Information visualization contains two fundamentally related aspects: *structural modeling*, in which data relationships are detected and extracted, and *graphical representation*, which translates the data structure into a visual representation. Within the broad field of information visualization, the research presented in this thesis constitutes an *exploratory data analysis* metaphor that facilitates a process of *visual information seeking* within *abstract data*.

2.2.1. Abstract Data

Information visualization faces the need to represent the structure of and the relationships within *abstract data*. Abstract data is characterized by its lack of a natural notion of position in space. Typical visualization examples that represent this data type include financial models, textual analysis, transaction data, network traffic simulations and digital libraries. Information visualization thus clearly differs from so-called *scientific visualization* of *physical data*, which implicitly carries the spatial layout that can be visualized by forms of graphical reproduction, such as geographic layouts, architectural plans, medical imaging or weather maps.

Because the data is non-spatial and lacks natural representation, the fundamental challenge for information visualization is thus "… how to invent new visual metaphors for presenting information and developing ways to manipulate these metaphors to make sense of the information" (Eick, 2001). Additionally, such data mapping algorithms need to be automated, hereby removing the effort of translating huge data quantities into a visual form by hand. Ideally, such visualization metaphors are robust and capable to represent all sorts of datasets even for non-expert users.

2.2.2. Information Exploration

In practice, *exploration* signifies the process of *browsing* through data. It is a highly interactive process, as the visualization itself has a wide influence on the next interaction step taken by the user. Formal studies have proven that visual exploration can support cognition because users can exploit their personalized domain knowledge to adjust goals automatically through an interactive interface (Larkin and Simon, 1987). The concept of exploration is especially useful when little is know about the data and the visualization goals are vague.

Exploratory data analysis or *information exploration* is a conceptual approach that employs mostly graphical techniques to maximize insight in the dataset and to test underlying assumptions. This philosophy does not consist of a set of known techniques, nor does it test assumed data models that follow from a statistical analysis. Instead, it focuses solely on an efficient visual analysis of the data structure itself, out of which an empiric model emerges (NIST/SEMATECH, 2003). In general, a classical analysis tests the variables of a certain predefined model, while exploration implies that the analysis attempts to detect the model that would be appropriate.

- **Classical Analysis Sequence.**
  Problem - Data - Model - Analysis - Conclusions
- **Exploratory Data Analysis Sequence.**
  Problem - Data - Analysis - Model - Conclusions

Upson et al. (1989) describe data exploration from the perspective of the visualization instead of the user. For them, the visualization process is an *iterative analysis cycle*. As shown in Figure 2.1, a dataset is filtered into subsets of interest, mapped onto visual primitives and then rendered for the user by a function called the visualization transform.

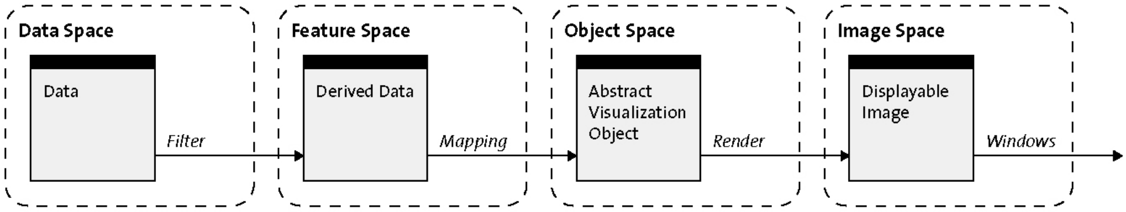


Figure 2.1. Dataflow model (Upson, et al., 1989)

The representation generated by this conversion process is then used by the user to provide feedback on the previous steps, thus restarting the cycle. The key feature of such a model is that the visualization process is an iterative sequence of user controlled transformations. Consequently, elements that change during this iteration must be the focus of any description of the visualization. These process descriptions are particularly important when visualizing and manipulating time-dependent data flows, as fresh data continuously enters the visualization space and requires constant reevaluation (Jankun-Kelly, et al., 2002).

2.2.3. Visual Information Seeking

Visual information seeking combines the concept of visual information presentation with dynamic user control techniques. This approach differs drastically from the concepts of classical query composition or *information retrieval*. Figure 2.2 demonstrates how the latter process is restricted to searching and querying data to retrieve specific data values. Information seeking, shown in Figure 2.3, focuses instead on rapid filtering techniques to reduce the resulting datasets and progressively refine search parameters.

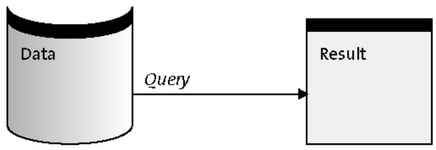


Figure 2.2. Classical information retrieval.

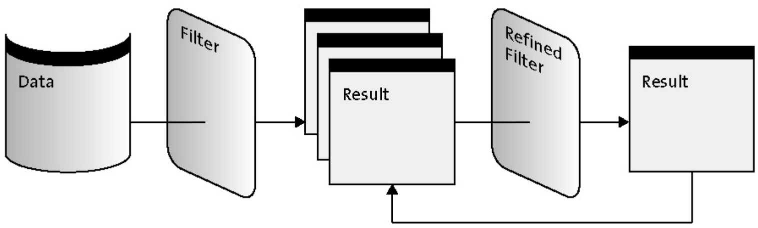


Figure 2.3. Visual information seeking.

Users are able to visually identify intermediate results, and are encouraged to reformulate their goals during the data exploration process itself. Shneiderman's (1996) so-called *visual information seeking mantra* implies the use of direct-manipulation interfaces, and defines the sequence as: overview (a collection) first, zoom in (on items of interest) and filter (uninteresting items), then show details on demand (of a single item or a group of items).

2.2.4. Related Work

The field of information visualization comprises many different data representation techniques. The following sections describe working applications that show conceptual similarities with the infoticle approach. Notably, information visualization techniques that are based upon virtual reality technology or particle systems are not listed here, but in the subsequent sections.

2.2.4.1. Scatterplots

Scatterplot representations like so-called Starfield Displays (Shneiderman, 1994) demonstrate the use of hundreds or thousands of spatially distributed points organized in static graphs. They are used to visualize multi-dimensional datasets, are available in many variations, such as cone-trees and hyperstructures, and have already been applied in virtual reality systems (Ebert, et al., 1996, Lamm, et al., 1996, Jaswal, 1997, Nagel and Granum, 2002).

Scatterplots are able to handle large amounts of data in an efficient manner by translating numeric data values into spatial positions, typically mapping data values onto singular spatial coordinates. The resulting clusters of points are then used to interpret the information, a method also used by the infoticle metaphor.

2.2.4.2. Force-Placement

The use of force-placement and spring-embedded algorithms is a widely investigated topic. An in-depth discussion on force-directed graph layout algorithms along with information on state-of-the-art graph drawing can be found in (Chen, 1999).

Spring-embedded methods are capable of generating so-called *undirected graphs*. The development of these graphs can be traced back to a Very Large Scale Integration (VLSI) design technique, which attempts to optimize the circuit layout with a minimal number of line crossings. The spring-embedded model was originally proposed by Eades (1984), using both attractive and repulsive forces to generate uniform edge lengths and symmetry. Practically, graph vertices are replaced with steel rings, and each edge is represented by a spring. The spring system starts with a chaotic stage, and achieves an optimal layout when the system energy is reduced to a minimum. In most force-directed visualization systems, spring stiffnesses correspond to certain pre-computed similarity measures, out of which an energy minimum is calculated that reveals multi-dimensional relations and adjacencies in terms of spatial neighborhoods.

Current research in this area focuses on the development of efficient algorithms that can handle large amounts of nodes or deals with user interface issues, the occlusion of nodes, and the use of virtual reality (see also Section 2.3.4. Background – Virtual Reality – Related Work). The force-placement methodology also has been combined with the generation of blobby surfaces that span the resulting point collection, alike the infoticle shapes (Gross, et al., 1997, Sprenger, et al., 1998).

The force-placement technique shows some conceptual similarities with the infoticle metaphor. As an infoticle constellation, a spring system requires a certain time span to reach a true representation, and uses the concept of attraction and repulsion to generate clusters of similar data objects. However, there are some fundamental differences between both the approaches as well: infoticles do not require a specific dedicated calculation time to generate the scene, nor retrieve data values from predefined similarity matrices to determine the attraction strengths. One should also note that the final visual appearance of a force-directed representation is not reproducible, either because of the randomness of the initial node positions or because of major constellation changes resulting from small layout parameter adjustments.

2.2.4.3. Time-Varying Data

To date, the few time-varying dataset visualizations applications that can be found in literature often seem experimental and are often static in nature. However, time simulation is a critical feature for many datasets, as it enables sudden event tracking or trend detection.

Many approaches exist that map time onto spatial lines, also called *timelines*, such as piles (Mander, et al., 1992), LifeStreams (Freeman and Gelernter, 1996) and LifeLines (Plaisant, et al., 1996). Similarly, ThemeRiver represents temporal thematic variations within a large collection of documents using the metaphor of a flowing river (Havre, et al., 2002). Notably, all these timeline representations are static.

The continuous feed of news articles is a typical form of dynamic, real-time data, which has been visualized as changing attributes of a four-dimensional landscape or so-called TextScape (Rossi and Varga, 1999). The dynamic aspect is conveyed to users by rebuilding and subsequently switching the whole scene when new data arrives. Most other news feed visualization approaches have been accomplished with little consideration of representing the time-varying aspect (Earl, 1994, Sparacino, et al., 1999).

The Control Project exploits the concept of flow for visualizing adaptive database query streams (Hellerstein, et al., 1999). *Control applications* (like online aggregation) can be classified as *anytime algorithms*, i.e. algorithms that can produce a meaningful approximate result at any time during their execution. Instead of requiring a dedicated timespan to generate the scene, the Control Project gradually streams representative data objects to the visualization system. In some respect, this approach is similar to the infoticle method, as both provide the visualization system with large remote datasets and require a certain time span to reach a true representation.

Only few approaches currently exist that merge temporal data with virtual reality technology. The so-called 3-D Visual Data Mining system (3DVDM) is especially adapted for dynamic datasets, generating three-dimensional scatterplots that morph between two static states (Nagel and Granum, 2002). Similarly to the infoticle approach, this method solves time-varying issues by distributed computing and shared memory parallelism. Motion is used to denote change from one time state to another, next to the use of *point vibration* as informational cue, in terms of amplitude, frequency and phase in all three dimensions.

A more theoretical evaluation of currently existing time-varying information visualization approaches can be found in Section 5.2.1. Analysis – Evaluation – Methodology.

2.2.4.4. Shape

Rohrer and Sibert (1998) use implicit surface models, so-called *concept shapes* or *blobby texts*, which span a spring-embedded representation in order to map document content onto shapes. Similarly to the infoticle shape generation, generalized, procedurally-generated density fields and implicit surface models are used to create the three-dimensional shapes. However, the document comparisons are static and are retrieved from a pre-computed numerical similarity matrix.

Stereo Field Analyzer (SFA) shows a clustered document space as a three-dimensional scatterplot with stereoscopic capabilities (Ebert, et al., 1996). This method merges a traditional three-dimensional scatterplot representation with *procedurally-generated shape* differences of superquadric forms, such as roundness and pointiness variations, in order to visualize extra data dimensions (Shaw, et al., 1998).

The method of flowing particles that follow perceptual and physical principles along the surface is employed as a novel visualization technique, called *kinetic visualization*, to aid the spatial perception of static shapes (Lum, et al., 2003).

2.2.4.5. Motion

Robertson (1993) describes in general terms the principles of three-dimensional animation and distortion techniques for information visualization.

Magnitudes and load currents are represented as animated vectors for visualizing the dynamic behavior of electric power systems, exploiting visual coherency to recognize groups of machines *swinging* together in time (Gronquist, et al., 1996).

Motion is mostly used as a process of *morphing*, in which a smooth blending is generated between two distinct graphical objects. Various morphing animations construct a multi-dimensional representation space that can be used for visualizing multi-dimensional data. In this context, Mona Lisa faces or hand representations can be smoothly morphed between two fixed graphical states to denote various abstract data attributes (Müller and Alexa, 1998). Discovery Visualization is an approach that uses a constant unfolding of data and the positional morphing of data points to represent time-dependent datasets (Ribarsky, et al., 1999). It uses fast clustering methods to reevaluate and visualize the time-varying data as a continuous animation simulation.

Wave Motion is an original information visualization approach, revealing data patterns by interferences between waves (Azzopardi, 1999). Just as stones dropped into water generate rings of ripples spreading outward to interact with each other, data points generate waves with a particular wave length, frequency, phase and amplitude. Although the current state shows emerging visual patterns that seem very difficult to interpret, theoretically even time-varying datasets could be visualized.

Motion can also be used as vibration, in terms of frequency, amplitude and phase to denote data dimensionality in the context of virtual reality and time-varying datasets (Nagel and Granum, 2002), or information visualization perception experiments (Bartram and Ware, 2002).

2.2.4.6. Financial

Physically-based force model layouts, such as the Bead visualization method, have been adapted to the needs of the business environment (Brodbeck, et al., 1997). Notably, conservatism and skepticism with regard to three-dimensional visualizations led to the development of a fundamentally two-dimensional tool. Several examples of current state-of-the-art approaches in the area of financial visualizations for consultancy businesses can be found in (Chuah, 2002) and (Taskaya and Ahmad, 2003).

2.2.4.7. Knowledge Network

Many aspects of electronic networks can be visualized: the structure of the network, the geographic connectivity, optimal pathways and so on. Only few scientific approaches can be found that focus on geographical usage network or, like the infoticle method, on the accurate visualization of dynamic access patterns. Notably, most network visualizations represent the physical transmission of network packets, whereas galaxy infoticles represent the more abstract usage of equal document contents.

The SeeNet network visualization system employs animation to scan data from many time periods (Becker, et al., 1995). The Avatar virtual reality system analyzes and visualizes geographical and temporal patterns within web logs in real time (Lamm, et al., 1996). Basically, so-called Scattercubes, three-dimensional generalizations of two-dimensional scatterplots, are generated, supporting the analysis of very high-dimensional, non-grid based, time-varying data.

Typical e-business data visualization focuses on problems of scale, dimensional complexity, data analysis and interaction tasks when representing the paths (traces of visited pages) and flows (aggregations of paths) of users visiting websites (Eick, 2001). Tudumi is an original three-dimensional method for visualizing huge web logs from the perspective of network administration and intrusion detection (Takada and Koike, 2002). Many other approaches are restricted to traditional two-dimensional interfaces and data mining tasks (Sifer, 2002). VISIP visualizes the paths taken through a website by generating curved lines, conceptually resembling infoticle splines, but here simply connecting static diagrammatic elements (Cugini and Scholtz, 1999).

A more artistically inspired approach uses gravitational forces inside a three-dimensional environment to visualize web log data (Wakita and Matsumoto, 2003). In contrast, the Cybernet project is a comprehensive research project that visualizes network data for network monitoring and management by constructing real-world mimicking three-dimensional virtual worlds that resemble a city or a solar system (Russo Dos Santos, et al., 2000). Special care was taken to convey a consistent, real world based metaphor, so that e.g. the proportions of buildings would be believable and not become too stretched or flat by the automatic mapping algorithms. Cybernet represents dynamic data updates by smooth animations between separate static states.

2.3. Virtual Reality

Maybe inventing virtual reality in human history, Sutherland (1968) specified a sophisticated hardware system that is worn in front of the eyes, tracks the motion of the head and creates a compelling, three-dimensional experience for the user. In these early days of computing, computers still being more like machines, huge and expensive, Sutherland imagined a room in which the computer could control the existence of matter, and concluded that, with appropriate programming, such a display could literally be the "Wonderland in which Alice walked". In 1989, Jaron Lanier coined the term *virtual reality* in an attempt to encompass all of the virtual projects then being investigated.

Currently, virtual reality denotes the technology used to simulate some kind of believable actuality through the manipulation of sensory feedback using the latest electronic and digital developments. It acts like a technological tool that provides an intimate interface between humans and computer imagery, enabling users to interact intuitively with fully convincing, computer-generated, three-dimensional environments. These virtual worlds can be whatever the designer makes them, deliver intimate and powerful experiences of new ways of exploration and enable any action imaginable. Notably, these capabilities do indeed hold great promise, but they still imply a major challenge: designing and orchestrating meaningful action in virtual worlds. Because virtual reality is a relative new medium, its theoretical definition is still in flux, although most authors define its core characteristics as the combination of a virtual world, immersion, sensory feedback and interactivity (Sherman and Craig, 2003).

Current virtual reality research focuses on advances both in hardware as well as in software, with the primary goal of increasing the degree of realism and complexity, such that the virtual world becomes more believable and the degree of immersion is enhanced. Researchers are continuously developing new hardware technology that allows for the rendering of more complex three-dimensional scenes at interactive framerates, and typically deals with advances in graphics display, tracking or sound technologies. Simultaneously, software algorithms have been implemented and refined in the areas of model simplification, level-of-detail culling, texture mapping, lighting and shading, hidden surface elimination and so on.

2.3.1. CAVE

The research presented here is primarily based upon the qualities and characteristics of *immersive virtual reality* (IVR), also labeled as *spatially immersive displays* (SIDs). The most popular hardware installation in this realm is called the Cave Automated Virtual Environment (CAVE) (Cruz-Neira, et al., 1993). Such an environment includes three or more planar display screens that are positioned in a room-sized cube-like setup completely surrounding the user, and is coupled with a head tracking system to produce the correct stereo perspective. The viewer explores the virtual world by moving around inside the cube and grabbing objects with a three-button, wand-like input device (see Section 3.6.3. Infoticle – Interface – Interaction). This sophisticated display technology offers the four main characteristics of virtual reality.

- A **wide angle of view** to immerse participants in human scale images.
- **Stereoscopic vision**, which creates and displays a different image for each eye.
- **Interaction**, to enable participants to move within the environment and to change it.
- **Viewer-centered perspective**, calculated in real time from a participant's point of view.

CAVE installations differ in terms of the number of walls on which the virtual world is projected. A classic CAVE has three or four (including a ground projection) walls, which are about 2.3m by 3m in size. A multi-processor computer equipped with powerful graphics hardware guarantees a synced rendering over all walls and a seamless user perception of the real-time, stereo computer-generated images that typically depict a three-dimensional world. Rear projectors avoid the casting of shadows on the walls generated by users, while a continuous spatial sound simulation enhances the illusion of immersion. Virtual reality users wear stereo glasses, and the orientation and position of their head is tracked, in order to create a *viewer-centered perspective*. In addition, a joystick or wand is used as the primary input device that enables users to navigate inside or manipulate the virtual world.

2.3.2. Immersion

A computer-generated environment is depicted as *immersive* when it appears to enclose the user, and when parts of the physical world that are not integral system components are blocked from view. Virtual reality technology typically immerses a user *physically*, in contrast to most media creators who try to immerse users *mentally*. Immersion thus can also denote the mental state of mind being fully occupied with a certain subject (see also Section 1.1.6. Introduction – Motivation – User Engagement). Technically, virtual reality experiences can be presented to users through different sorts of interfaces, which each result in a different degree of immersion.

- **Desktop Systems.** Navigation through a three-dimensional world on a monitor.
- **Partial Immersion Systems.** Navigating through a three-dimensional world on a monitor with enhancements such as gloves and stereoscopic goggles.
- **Full Immersion Systems.** The combination of head gear, gloves and bodysuits.
- **Environmental Systems.** Externally generated three-dimensional worlds, but with little or no body paraphernalia.

Telepresence employs virtual reality technology such as cameras and microphones to replace the corresponding senses of a participant. **Tele-immersion** in turn utilizes technology closely related to that of virtual reality, to facilitate the seamless integration of human participants within the virtual environment in real time, thereby drastically enhancing collaboration scenarios. This concept brings remote participants together inside the computer-generated world by recording and constructing three-dimensional representations of the users in real time. However, the integration of real-time three-dimensional video acquisition and reconstruction while simultaneously displaying the computer-generated virtual world to the recorded participants poses great technical challenges. At present, both The National Tele-Immersion Initiative (Sadagic, et al., 2001) and the interdisciplinary blue-c project at ETH-Zurich (Gross, et al., 2003) constitute the most comprehensive projects that attempt to address all these issues. Possible virtual reality applications based upon the concept of tele-immersion include new shopping experiences (Lang, et al., 2003), car manufacturers design reviews, virtual exhibitions and environmental or health care data visualizations (Leigh, et al., 1999c).

The work presented here was performed within the context of the blue-c project framework at ETH-Zurich, which has developed and implemented a CAVE-like tele-immersive virtual reality theatre, and has built another networked virtual reality installation at a physically remote place (see also Appendix A). In this thesis, the term virtual reality will thus be used to denote technology that facilitates an immersive, stereoscopic experience of computer-generated worlds.

2.3.3. Application

Pushed by the success of game consoles, entertaining immersive experiences (Mine, 2003) and e-commerce scenarios (Mass and Herzberg, 1999), an increasing amount of virtual reality research focuses on developing user-friendly hardware prototypes and useable applications that introduce high-tech display and interaction technology to a wide audience. In fact, virtual reality technology has advanced to the point that it is currently possible to develop and employ meaningful, productive applications. Notably, only a few immersive virtual reality applications have become successful in terms of common use outside the laboratory. These applications have been proven to effectively fulfill their purpose, or reduce financial costs and time for their users (Bowman, 1999).

- **Architectural walk-troughs** or **simulations** are used to immerse users in a three-dimensional model of architectural space to explore the visual impact and physical aspects from a first-person perspective (Schmitt, et al., 1995).
- **Psychotherapy experiments** use virtual reality technology to confront patients with a convincing virtual world that provokes mild forms of their phobias for therapeutic purposes.
- **Games**, typically first-person shooting situations, bring together physically remote participants for entertainment purposes.

Other immersive virtual reality applications have received less mainstream use, most probably because they require more complex forms of interaction, such as navigation, object selection and manipulation, system mode changes, and real-time simulation adaptations. These applications, situated in fields such as flight and vehicle simulation, design modeling and scientific visualizations, typically fulfill highly specific and specialized goals and are mostly developed for experts rather than for the general public. Notably, the relatively small number of newly developed applications and the general low acceptance rate of virtual reality technology are also caused by the fact that applications still have to be programmed by skilled and dedicated programmers, as user-friendly virtual reality authoring tools are still unavailable and thus a thorough knowledge of a low-level programming language is required. Currently, the creation of a virtual reality experience even requires a reasonably large team effort, as it relies on the combined use of complicated input and output devices, rendering hardware and specialized software layers.

2.3.4. Related Work

Many traditional visualization methods have been ported to virtual reality technology, yet only a few have been specially *invented* to be employed on this platform. However, the combination of information visualization and immersive virtual environments are creating a number of novel application possibilities (Grady, et al., 1998). The following visualization approaches were initially designed to be used within virtual reality environments, so that most of the corresponding metaphors are based upon three-dimensional virtual worlds, and have specifically designed navigation paradigms.

The very first approaches to use three-dimensional space for browsing and locating information were inspired by the notion of *spatial databases* (Fields and Negroponte, 1977). In such virtual worlds, database objects can be distributed in meaningful locations, so that the location of a viewpoint becomes a *query*, and *browsing* resembles information spaceflights (Caplinger, 1986). As one of the first fields of virtual reality application, the increasing computational power facilitated the astrophysical exploration of the vast universe dimensionalities *from within* (Deyang and Norman, 1994).

Bead is based on the metaphor of information landscape, mapping interrelationships among a set of documents onto a geographical-like model within the DIVE virtual reality environment (Chalmers and Chitson, 1992). Because of negative experiences with the visual complexity of and navigation within three-dimensional models, Bead restricts the range of fluctuation in the third dimension.

Undirected graphs are used in virtual reality applications, such as Q-SPACE (Pettifer, et al., 2001) and VR-VIBE (Benford, et al., 1995). Q-SPACE encloses regions in hulls, or so-called DataClouds, to support the recognition of data clusters. Special attention was given to aesthetics, real-time direct interaction, spatial awareness and potential multi-user capabilities. VR-VIBE is a virtual reality based system, representing interrelationships among documents in response to queries based on user-specified keywords. Both keywords and documents are placed in the three-dimensional space, and distances between them denote the number of keyword matches within the documents. VR-VIBE also enables multiple users to browse the same scene.

Virtual reality research also focused its attention on connecting remote sites and offering distributed access to datasets, independent of the respective immersive capabilities (Reed, et al., 1997). Novel virtual reality environments have been implemented to answer the need for a new generation of data access, data mining, data visualization and networking tools that are able to handle the growing requirements of data handling and querying. The Tele-Immersive Data Exploration Environment (TIDE) is a collaborative, networked virtual environment, which enables the exploration of massive datasets (Leigh, et al., 1999b, Sawant, et al., 2000). On a smaller scale, CAVERN, the CAVE Research Network, interconnects pairs of immersive virtual reality installations to support collaboration in various disciplines, such as design and visualization (Leigh, 1997). Immersed Virtual Environment Over a Network (DIVE-ON) uses the concepts of virtual reality, databases and distributed computing to experiment with new approaches to visual data mining (Ammoura, 2001). Therefore, a typical Database Management System (DBMS) is transformed to construct data cubes using spatial coordinates, color and scale as the main visual cues.

van Dam et al. (2000) extensively analyzed the current challenges of immersive virtual reality and scientific visualizations. They even propose a whole new design discipline and validation methodology for this fundamentally different medium and call for proving immersion's effectiveness for scientific applications and tasks, as only a few quantified user studies in this area exist.

A multitude of tele-immersive applications have been evaluated by Leigh et al. (1999c), resulting in a set of practical rules-of-thumb for developing applications in this realm. Some approaches use so-called *minimally immersive interfaces* consisting of a normal, stereoscopic desktop display and two three-dimensional trackers to recognize shape variations of volumetric elements within scatterplots (Shaw, et al., 1999). Others simply attempt to port the well-known three-dimensional pie charts, bar graphics and filing cabinet drawers to the world of virtual reality (Kirner and Martins, 2000). Imsovision visualizes object-oriented software by generating a VRML representation, which is shown in an immersive virtual reality environment (Maletic, et al., 2001). Sequence World uses simple position, color and shape cues to visualize small subsets of DNA sequences as text-labeled cubes on a navigable two-dimensional plane (Rojdestvenski, et al., 2000). In fact, the analysis of genome sequences has recently received attention from various researchers (Kano, et al., 2002, Adams, et al., 2002). In a more playful way, the Museum of Color merges immersive virtual reality with architectural insights to visualize different abstract concepts, such as color theory, for educational purposes (Palter, et al., 2000).

Notably, most of the previously mentioned information visualizations that operate on real-world datasets tend to require an off-line pre-computation to generate the virtual environment. As a result, the actual virtual reality experience of viewing the data is one of passively navigating a static pre-computed form. Others allow for dynamic interaction, but work with small datasets or utilize low-level placement routines that require little computation and typically produce little insight in the data. Typical real-time implementation problems include database communication, force-placement calculations, hull generation and fast rendering. Notably, except for the 3DVDM system, explained in more detail in Section 2.2.4., no other approaches exist that merge time-varying data visualization with virtual reality technology, let alone exploit this technology's unique perceptual and interactive characteristics.

2.4. Particle

The particle system functionality is explained in detail in Section 3.1.2. Infoticle – Basis – Particle. As particles have been used in a broad spectrum of applications, the following section focuses solely on those approaches that combine particles with abstract data visualization or show features similar to those of the infoticle methodology.

2.4.1. Abstract

Since most information visualization metaphors still use static data mapping techniques, the potential of particle animations in this field is poorly explored.

Particle trajectories influenced by gravitational forces have been represented in virtual reality environments to describe the general theory of relativity, showing the geometry of space-time using techniques taken from geodesics (Bryson, 1992). Particles and virtual reality have been combined to create a so-called Cosmic Explorer, which visualizes numerical and observational cosmology data (Song and Norman, 1993).

Particles trajectories have been used in combination with information visualization techniques to visualize real particle collision datasets from high energy experiments, which are helpful to physicists in investigating the behavior of interacting matter at high energy densities (Wei, et al., 2001).

The Spray Rendering technique uses the metaphor of *spray cans* to paint and visualize datasets (Pang and Smith, 1993, Pang, 1994). Conceptually, users grab and aim spray cans into their datasets in which, depending on the type of the can, different data types are highlighted and rendered. Spray cans thus use data values rather than color to highlight specific parameters, so that the paint particles can be imagined as *intelligent agents* that look inside the data space for targets and highlight features of interest. This visualization approach is similar to the infoticle method as it combines particle systems and behavioral animation to generate data patterns, however requires scientific iso-surfaces as the visual basis of the highlighting procedures.

Particle motions can also be applied for recognizing shape patterns in relationship with abstract, external influences. For instance, particles have been used to debug visualization systems, by representing geometry, attributes and relationships graphically instead of algorithmically (Crossno and Angel, 1999). This method employs the animation of colored particles to represent successive program iterations, so that visual pattern recognition skills can be applied to debug eventual visualization system problems. In a more general way, particle motions have also been applied to support the perception of shape (Lum, et al., 2003).

2.4.2. Boids

The boid methodology is explained in Section 3.5.4. Infoticle – Simulation – Behavior. The field of application of boids mainly focuses on the simulation of physical phenomena, such as the perception, intention, behavior and locomotion of swarming animals. Although boids are thus used in many contexts (Reynolds, 2001), their capabilities for information visualization has rarely been investigated.

The clustering movements of swarming fish inside a three-dimensional, virtual world can be used to visualize the relationships of interest among employees (Proctor and Winter, 1998). This visualization method is very similar to boid infoticles and is therefore explained in more detail in Section 3.5.4. The boid concept can also be used in a more technical context, as it is able to visualize and analyze the performance of network communications by characterizing different sorts of swarm behavior (Kadrovach and Lamont, 2002). This method developed a realistic mathematical model of dynamic formations in order to characterize swarms with respect to implementation parameters of wireless, ad-hoc network communications systems. The emergent behavior of local flock interaction can also be used to form an effective search strategy for performing an exploratory geographical analysis, much like detecting visual clusters in large collections of points (Macgill and Openshaw, 1998). It is an example of the so-called Particle-Swarm Optimization (PSO) research field that, like other evolutionary computation algorithms, can be applied to solve most optimization problems in the fields of system design, pattern recognition, biological system modeling, signal processing, decision making, simulation and identification, and so on (Eberhart and Shi, 2001).

2.4.3. Scientific

Particles are also used in scientific visualizations, mostly to represent real physical processes. *Computational Fluid Dynamics* (CFD) is a well-defined scientific field that models physical fluid phenomena that cannot be easily simulated with experiments because of physical, financial or time constraints (Merzkirch, 1987). CFD is an interdisciplinary science, combining knowledge of physics, applied mathematics and computer science. CFD methods predict quantitatively what will happen when fluids flow by solving science-based mathematical equations, using data about the circumstances in question. A *fluid* is a continuum consisting of many deformable particles (or drops), and is characterized by certain macroscopic properties (e.g. density, viscosity, pressure, temperature, velocity) which vary in space and time and change under the influence of external forces.

Both CFD and the infoticle methodology try to visually detect zones that represent dynamic behavior according to time-varying characteristics, such as steady versus unsteady, laminar versus turbulent. Similarly to the infoticle metaphor, CFD researchers describe fluid motion characteristics by animation, showing moving particles, or statically, by tracking individual particles as they move through the flow field (Hin and Post, 1993, Rosenblum and Post, 1993). Infoticle timeline ribbons are comparable to stream surfaces (Hultquist, 1992), or *stream polygons* (Schroeder, et al., 1991), as explained in Section 3.4.4. *Virtual windtunnels* are typical scientific particle visualizations that are displayed in virtual environments and can be manipulated in real time (Bryson and Levit, 1991, Kuester, et al., 2001).

Although infoticles might convey visual similarity with particles used in CFD visualizations, they are fundamentally different. Primarily, infoticles are not injected into flow fields, but instead exist in an empty, gravitation-less virtual space. In contrast, CFD particles are subjected to the mathematical simulation of statistically generated mean velocity and turbulence intensity fields. The only possible elements influencing infoticles, in contrast, are macroscopic point forces and internal behavior rules that are controlled either by pair-wise dependencies between infoticles themselves or by continuous data updates. This implies that the movements of infoticles during a simulation cannot be predicted, as there is no mathematically derivable relationship between the spatial regions through which infoticles pass.

As both scientific and infoticle visualization methods deal with the continuous spatial movement of mathematical points, some similarities can be found as well. As with infoticles, the visualization of three-dimensional, unsteady flow has to deal with a lot of perceptual issues, such as the occlusion of distant details, lack of directional and depth hints, and cluttering (Fuhrmann and Gröller, 1998). Both techniques need to continually update the position of all the particles with every animation step, causing considerable consumption of computing power. Both approaches also rely on stereoscopic cues, interactive and intuitive viewpoint changes and the feeling of immersion in order to avoid visual occlusions, and to allow users to get a better overall impression of the three-dimensional flow structure.

2.4.4. Graphics

Particle systems are a technique within the field of computer graphics for creating a wide range of complex visual effects, such as rain, explosion, waterfalls, fire and smoke (Reeves, 1983), and are typically used in games or to create movie effects. In this field, the term *particle system* has also been used to describe different modeling techniques, rendering methods and animation types. For example, particles can be used to model and reconstruct surfaces and volumes, and to simulate the physics of different kinds of materials. Accordingly, Tonnesen (1998) combines particles with self-organizing principles to create so-called *dynamically coupled particle systems* that generate dynamically changing structures for free-form shape modeling, computer-assisted animation and surface reconstruction. Particles can also be employed to generate and control three-dimensional surfaces, by specifying control points for direct manipulation of certain particle motions, then solving the surface motion that maintains these constraints (Witkin and Heckbert, 1994). The field of computer graphics attempts to extend and optimize particle algorithms (Ilmonen and Kontkanen, 2003), or particle tracing methods (Kenwright and Lane, 1996).

3. Infoticle

The term *infoticle* merges the concepts of *info*-rmation handling and par-*ticle* simulation, and is a synonym for a *data-driven particle*. In effect, the word infoticle denotes the very core of the visualization metaphor: dynamically mapping abstract data values onto attributes of individual particles, so that data update characteristics are represented by emergent visual patterns, such as dynamic behavior typologies, spatial clusters and three-dimensional shapes.

This chapter describes in detail the *design rationale* of the infoticle metaphor, and starts with listing the inspirations taken from particle-like phenomena originating from various scientific fields, resulting in the very first infoticle idea sketches. Subsequently, it explains findings taken from cognitive science that are relevant for the design and implementation of information visualization applications in general, and for the infoticle metaphor in particular. In addition, it is clarified how these tools, interface elements and behavior rules collectively compose the core of the infoticle metaphor.

3.1. Basis

Because the data is non-spatial and lacks a natural representation, the fundamental challenge of information visualization is the invention of suitable, automated visualization metaphors that enable users to make sense of abstract datasets and facilitate navigation through it. Traditionally, informational values are directly mapped onto abstract attributes of objects following a certain interpretation scheme.

3.1.1. Design

The evaluation of related work listed in Chapter 2 has shown that contemporary information visualization applications focus either on specialized standalone methods or on the combination of several known techniques that are specifically developed for specific dataset characteristics. Generally, the design, development and evaluation of a novel visualization metaphor rely upon research findings taken from diverse scientific areas, taking into account a whole range of usability and usage issues.

Accordingly, the following *technical design rationale* was considered during the infoticle method development.

- **Cognition & Perception.** The effectiveness of a proposed information visualization method is ensured when its mechanisms are compared with the human sensory and cognitive capabilities that are required to comprehend and manipulate the resulting data representation. In case of immersive virtual reality, this consideration must be complemented with various perceptual issues, such as the feeling of presence, spatial awareness and stereoscopic vision. As the infoticle method exploits differences in motion typology as a visualization cue, scientific findings of behavior perception and motion cognition should be respected as well.
- **Presentation Medium.** Application designers should be aware of both the technical restrictions and unique display capabilities of the targeted presentation medium technology. Typical issues that determine the design of virtual reality applications include interaction devices, display resolution, display size and the ability to generate a stereoscopic projection.
- **Computing Performance.** Although hardware technology still evolves at an exponential pace, *demo-oriented research* needs to consider the current limitations of computing performance. In effect, applications that rely on the combination of real-time visualization of time-varying datasets and direct user interaction cannot afford low frame refresh rates or slow response times, as these would considerably affect the user experience. Consequently, many computing aspects have to be optimized at the early design stage as well as during the implementation process itself.
- **Human-Computer Interaction.** The technical limitations and low bandwidth issues that are related to typical *human-computer interaction* influence the design process of virtual reality applications considerably. As mouse or keyboard are replaced by less common input devices such as a six-dimensional wand, a joystick or a trackball, virtual reality application designers are expected to invent specially adapted interaction paradigms that facilitate intuitive interaction.
- **User Engagement.** Current developments in multimedia applications show that users not only perform a certain task with a fixed goal, but also like to experience an agreeable joy alongside it. Therefore, usability testing should not be limited to task metrics, but needs to include measurements of attention focus, curiosity and intrinsic interest. *Engaging* applications usually incorporate insights from cognitive psychophysics and are inspired by various artistic approaches.

This list of design criteria should not be considered as exhaustive, but illustrates the typical elements that determine novel visualization metaphors at an early design stage. In fact, experience has shown that these considerations need to be continuously evaluated during all application development stages. These limitations become especially important when designing applications for electronic media in which common standards are still unknown and hands-on experience is scarce. Typically, when such novel technologies emerge, a specific amount of dedicated experimentation is needed to evaluate novel possibilities through a process of natural selection and survival, after which a certain *good practice* becomes apparent. Immersive virtual reality is still in the middle of this evolutionary route as the technology consists of relatively new display and interaction hardware and the application development still requires a considerable amount of specialized resources and dedicated time.

3.1.2. Particle

Particle systems were first formally proposed by Reeves (Reeves, 1983, Reeves and Blau, 1985) as a rendering technique that is able to realistically simulate dynamic, natural phenomena, and enables the generation of structured models. A *particle* can be imagined as a mathematical object in three-dimensional space that generally contains, but is not limited to, a fixed set of attributes, such as position, velocity (speed and direction), acceleration, mass, color, lifespan, age, shape, size and transparency. A particle system consists of a collection of particles, each of which is influenced independently through time by a set of predefined conditions (Martin, 1999). Currently, particle systems are a widely used computer graphics technique, so that the elements and logic needed to implement a real-time particle system are well known (Sims, 1990, Lander, 1998, van der Burg, 2000, McAllister, 2000). Particle systems can be combined with external forces, which influence the movement and speed of each single particle according to the laws of Newtonian mechanics. This way, particles are able to convey different visual effects, ranging from gravity simulation over surface bouncing to fading flames. Given correct initial conditions, and combined with internal relationships or external forces, particle systems can be animated over time, conveying complex behavior (Tonnesen, 2001).

3.1.3. Metaphor

This thesis presents the infoticle method as a visualization *metaphor* rather than a *model*, to emphasize the open-endedness, incompleteness, and inconsistent validity of metaphoric comparisons versus the explicitness, comprehensiveness, and validity of a model. Models are designed to represent some target domain, whereas metaphors are clearly chosen or designed to invite comparisons and implications.

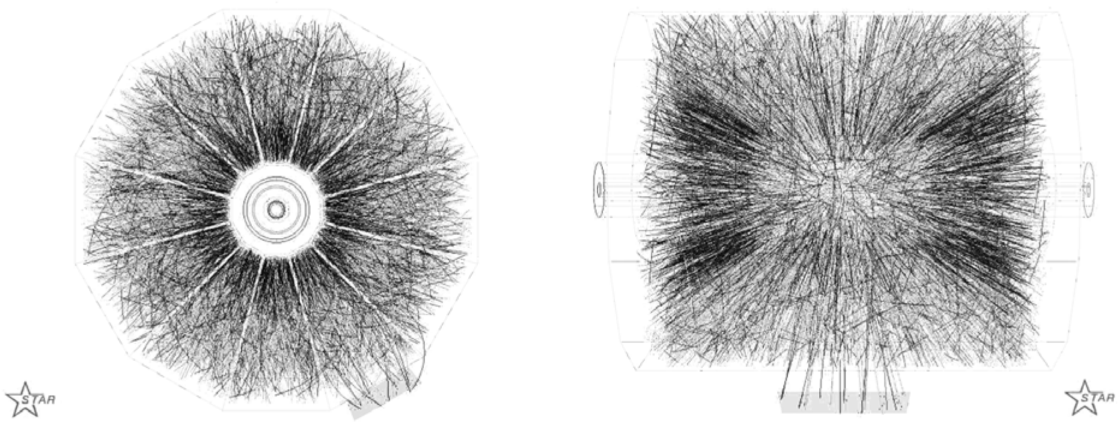


Figure 3.1. Subatomic collision representation (Brookhaven, 2003).

Particles, particle pathways and the interpretation of path representations have been used for many goals. Figure 3.1 shows the various particle trajectories that emerge during a subatomic collision created in a relativistic heavy ion collider. Although visually complex at first sight, physicists can make sense out of the directionality and clustering of these bunches of lines.

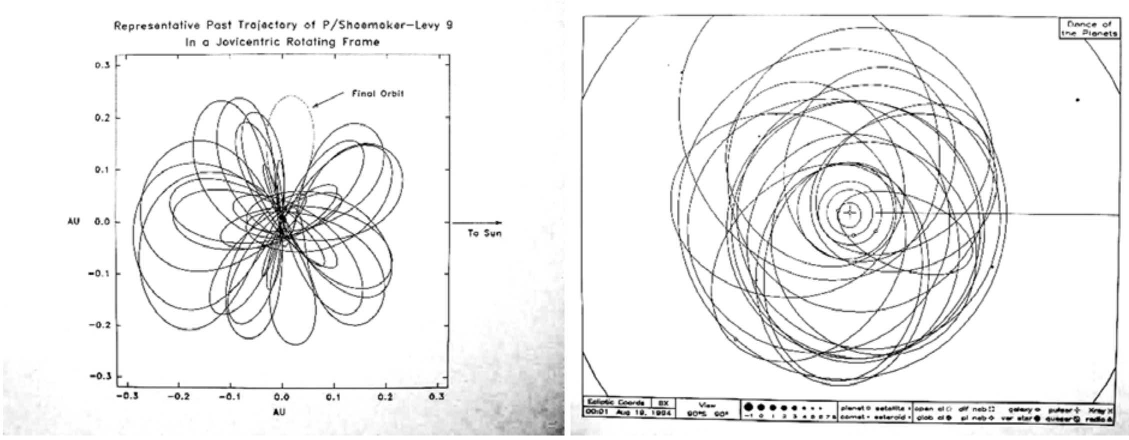


Figure 3.2. Comet trajectory path analysis (UCAR, 2000).

Figure 3.2 shows the trajectory of comet Shoemaker-Levy 9. Mathematical theory suggests that this was likely a short-period comet which was captured into orbit around Jupiter in 1929. Then, it began to execute a *jovicentric* path, ultimately ending with the collision on Jupiter. Accordingly, visual trajectories are used in scientific research to trace back and analyze unexpected events in history.

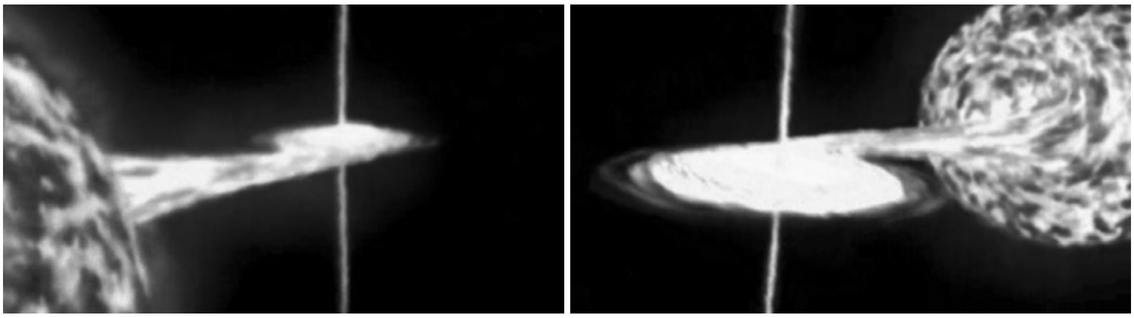


Figure 3.3. Star influenced by black hole (BBC, 2003).

Figure 3.3 shows an artist's impression of a black hole's influence on a nearby star. The black hole literally *eats* the star material during a period of millions of years, and bursts out all non-processed material in a straight line. Notably, although taking place in a three-dimensional world consisting of three-dimensional objects, the emergent visual pattern of the tremendous black hole *point attraction force* is a two-dimensional, flat disk, similar to some infoticle patterns (see e.g. Section 4.3.3. Application – Galaxy – Pattern).

Figure 3.4. Electron tracing (science_museum, 1997).

Figure 3.4 illustrates what happens when electrons traveling at near the speed of light collide with stationary electrons. Both these pictures are the result of studies of Arthur Compton in 1923, and international research accomplished in 1937. By measuring the *curvature* of the tracks, the energy of the electrons can be estimated. The shape and formality of simple lines thus are able to denote meaningful and informational values.

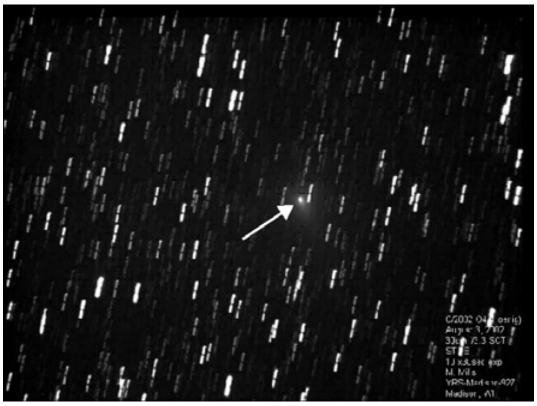


Figure 3.5. Comet detection.

Figure 3.5 in turn demonstrates the contemporary technique of comet detection. By comparing long exposures of the star sky, singular points *pop out* that do not follow the same trajectory or *behave* as the others. Similarly, information visualization can exploit this cognitively understandable phenomenon to visually represent outlying data entries embedded within large collections of common time-varying tendencies.

In contrast to most traditional information visualization approaches, information architects combine insights from artistic and architectural fields to represent information in such a way that it appeals to a multitude of cognitive senses in the background of human awareness. Next to the previously mentioned list of technical considerations, the infoticle method is based upon less measurable intentions that are mainly inspired by the concepts of information aesthetics and user engagement. Info-aesthetics (explained in Section 2.1.3.) deals with the creation and analysis of new media objects that visualize information processing. User engagement (described in Section 1.1.6.) reaches beyond typical usability issues such as task metrics and goal effectiveness, and relates to concepts such as enjoyment, curiosity and motivation.

Based on the theory of *metaphor use*, four steps can be identified when designing applications based on metaphors (Carroll, et al., 1990).

- **Identification of candidate metaphors**. Some general features should be present in the metaphor. The emotional tone of the metaphor must match the desired human attitude, the metaphor should describe a general theme, and be exciting.
- **Detailing of the metaphor analogies** with respect to representative user scenarios. This step describes the matches between specific user actions and the corresponding metaphorical events.
- **Identification of inevitable mismatches and their implications**. Evaluation of possible scenarios of use, prediction of shortcomings and implementation of possible alternatives.
- **Identification of design strategies** to help users manage mismatches. Possible wrong uses should be predicted, and error handling implemented.

In the early stages of the infoticle metaphor design, similar steps were evaluated, in particular to test the effectiveness of particle systems to visualize abstract, time-varying datasets and to facilitate intuitive interaction. In effect, the infoticle method is mainly based upon the visual qualities of particle systems because the apparent simplicity of dynamic particle movements holds great potential for representing complex systems. Animated particle systems are able to express multifaceted, but still interpretable behavior patterns that provoke human imagination.

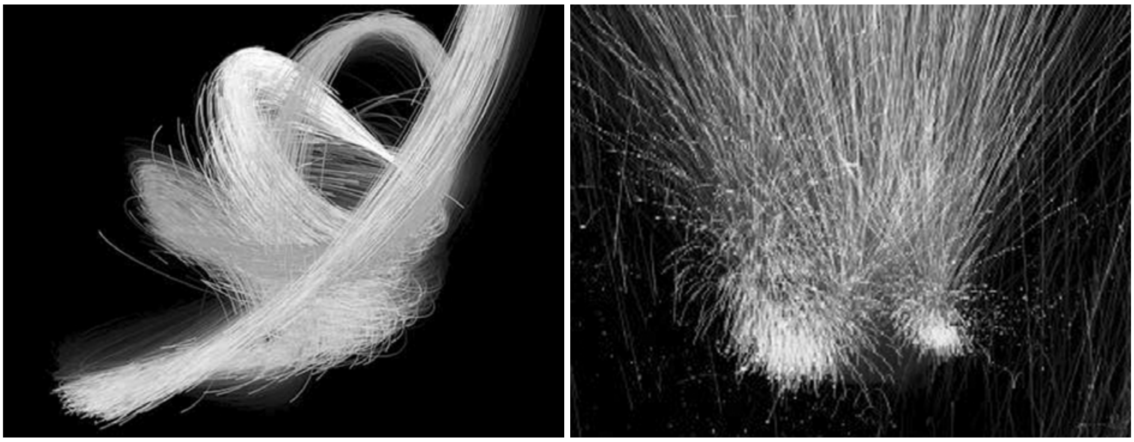


Figure 3.6. Dynamic particle patterns. (McAllister, 2001)

Figure 3.6 illustrates the visual qualities of constantly flowing particles that can be mentally linked to various characteristics of dynamic data streams. The combination of atomic point movements and global clustering characteristics of particle systems facilitates both a detailed analysis as well as a contextual overview. Aesthetically, carefully designed particle systems are able to convey visual effects that are enjoyable and pleasing to the eye. Particles exploit many effects of stereoscopic vision by enhancing the perception of *cue-of-motion* while simultaneously avoiding perceptual difficulties of spatial occlusion. The next sections will describe how particles convey the inherent capability to build up three-dimensional shapes using *implicit rendering* algorithms and are able to generate emergent spatial behavior by a process of *local interaction*.

The use of immersive virtual reality environments offers unique opportunities for metaphor designers to exploit the advantages of virtual worlds, especially in the direction of space perception and virtual interaction. Various concept sketches made at the early infoticle metaphor design stage demonstrate some of the capabilities that can be imagined only for computer-generated worlds perceived on human-scale displays. In particular, Figure 3.7 illustrates (a) how particle streams can be grasped and manipulated by simple hand movements, (b) how a user is able to build a personalized information environment by modeling a spatial constellation of human-size objects that influence particle streams and (c) how the manipulation of the world scale enables cluster analysis at various contextual distances.

Figure 3.7. Concept sketches.

3.2. Cognition

Partly physical, partly psychological, the fields of *psychophysics* or *cognitive psychology* denote all types of investigations into the nature of human sensory processes. These terms especially apply to the part of the vision process that is concerned with information transfer from the eyes to the effectors, either directly or through the medium of memory. The Gestalt School of Psychology was founded in 1912 to investigate how humans perceive *form*. Several theorists argued that a form is not perceived by summing up all its individual components, but as a coherent *Gestalt*. These researchers attempted to discover and describe the principles and mechanisms of perceptual visual processes that result into perceptual coherence, and synthesized them in seven different and still commonly used *Gestalt Laws*. Cognitive science represents a large collection of knowledge which has been gathered during many decades, consisting of various insights in how humans perceive, process and comprehend visual information. It is built upon the careful evaluation of empirical data, collected from scientific experiments that focus on specific human cognitive capabilities.

The development of novel information visualization metaphors should use insights from psychophysics to assure that design decisions are supported by scientific evidence. In fact, cognitive psychology is a rich source of both practical and theoretical information for the design of applications that utilize visual stimuli to communicate with users. Cognitive findings are also useful when the employed hardware technology is relatively unknown and its capabilities are still hardly explored, so that good-practice experience and commonly accepted design guidelines are not available. At least theoretically, insights from cognitive psychology lead to more intuitive comprehensions of visualizations by freeing up cognitive resources for other tasks, such as interaction, navigation and data exploration. The study of perception provides a better understanding of how information is perceived, how different visual types can be combined to present multi-dimensional data and how the effectiveness of visualization techniques can be assessed. Ultimately, the close integration of human cognition in visualization design should result in well-considered systems that facilitate the rapid and accurate exploration and analysis of large, complex, multi-dimensional datasets.

3.2.1. Stereo

The use of three-dimensional virtual worlds for visualizing abstract information is still a rather controversial representation technique (Shneiderman, 2003), not to mention the use of stereoscopic displays. Most critics argue that sophisticated virtual reality installations perceptually overwhelm users, and that many related contextual parameters render effective usability testing almost impossible. In effect, although most users seem to appreciate three-dimensional graphs, it is still unclear whether such displays improve performance, speed, accuracy or memory for data (Levy, et al., 1996). Recent research described a decrease in performance to locate items as the freedom in the third dimension increases (Cockburn and McKenzie, 2002), although others state that a three-dimensional visualization is the most effective when the interface is extensively adapted to both the user and the specific task (Sebrechts, et al., 1999).

The findings from cognitive science are particularly interesting for virtual reality technology, as it inherently employs stereoscopic capabilities. Nakayama and Silverman (1986) have shown that stereoscopic depth is the most powerful, pre-attentive visual feature as compared to, for example, color intensity and hue, two features often used in scientific visualizations. Moreover, they concluded that stereoscopic depth can be effectively used to overcome the effects of conjoin or visual occlusion. In effect, the rendering of large amounts of points on traditional displays typically results in a chaotic collage of overlapping, blended elements. Stereoscopic vision however, enables users to perceive the black gaps in the depth direction, dramatically increasing the virtual space that can be perceived. Experiments have shown that stereo increases the amount of points that can be effectively understood with a factor of three (Ware and Franck, 1996). Other research showed that, in an immersive virtual reality, users perform better on statistical data visualization tasks, such as cluster analysis, but worse on interaction effectiveness, than on a traditional desktop (Arns, et al., 1999). Search tasks, which are often required during explorative data analyses, are completed faster in immersive environments than on normal desktop machines (Pausch, et al., 1997).

Motion and stereoscopic perception are closely related, and have the capability to reinforce each other dramatically. In effect, motion provides an additional source of visual information about the spatial arrangements around us through the use of *depth cue-of-motion* or *structure-from-motion*. When an object or an observer moves in the physical world, the objects located at different distances move with different speeds and create patterns of motion parallax and kinetic depth that provide highly informative depth cues. This phenomenon enables a relatively accurate depth estimation of objects within three-dimensional space, even when these are displayed on traditional two-dimensional flat screens.

3.2.2. Motion

Motion is generated through a process called *animation*, in which form and structure evolve through a certain development over time, hereby conveying the feeling of *kinetics* or *dynamics*. Well-designed motion metaphors can be aesthetically appealing, and are able to attract attention and maintain motivation. Mostly, computer-generated motion is used to demonstrate change through a process of *morphing*, or to grasp the awareness of participants by displaying unexpected events.

The literature about *motion perception* is broad and well-established, dealing with various issues such as the representation and perception of motion (Badler and Tsotsos, 1986), the science of motion (Smith, et al., 1962) and motion and form (Kolers, 1972). Many authors assume that animated graphics are able to facilitate comprehension, learning, memory, communication and inference. Results from different psychological studies suggest that motion holds the promise of conveying *meaning* and thus is capable to describe how different elements are related. Nevertheless, motion is still an underutilized vehicle for displaying informational values, as most current visualization techniques still use static patterns, even to represent time-varying datasets. However, motion can be considered as a visual attribute of an object that represents information, and thus be treated along with other, more common visual cues, such as size, color and position.

Psychophysical evidence has shown that, within certain limits, humans tend to resolve local *motion ambiguity* following the Gestalt Law of *common fate*, which states that objects that move together in the same direction, become perceptually grouped (Wertheimer, 1923). Michotte (1963) discovered that humans perceive distinct points with equal speed and parallel directions as a single process, a phenomenon called *kinematic integration*. Ternus (1926) extended the discovery of Gestalt organizing principles to a characteristic called *apparent motion*. This term describes the generation of motion not by physically moving an object, but by successively appearing and disappearing equal elements that are displaced in space. A number of studies have shown that people can see relative motion with great sensitivity, rivaling the current, common use of static patterns for information display. For instance, Heider and Simmel (1944) showed that observers attribute high-level intentions and even emotions to movements of simple, geometric shapes.

The *perception of motion* depends on a number of factors, including displacement speed and acceleration, the color and shape of the elements, and the constellation of the other elements within the environment (Kramer and Yantis, 1997). *Temporal grouping* relates to the appearance and locations of the elements, the proximity in time and similarity of motions. The physical representations of object motions can be categorized in terms of their complexity (Proffit and Kaiser, 1995). For instance, *particle motion* is a class in which only the location of the object's center of mass is relevant to the dynamic behavior. In contrast, *extended-body motions* use mass and orientation in the mathematical equations of motion. Notably, scientific experiments have proven that people make poorer judgments in an extended-body context since these require interpretations of multiple natural dynamic object descriptors instead of one. Simply stated, people rather tend to consider only one thing at a time when making judgments about dynamic behavior.

One of the major concerns when designing dynamic representations is to create a global animation that is simultaneously clear and comprehensible, attractive and appealing. To be effective, motion should never become too complex, too busy or too distracting, either spatially or temporally. Motion has been researched in many application areas, from which many findings can be broadened to a more general use as well. Stasko (1993) derived four design guidelines from the principles of traditional animation: the *appropriateness* of process representation, *smoothness* and location, *duration* and control, and *moderation*. Similar empirical research focusing on the effectiveness of motion in teaching scenarios resulted in the following list of critical characteristics (Hansen, et al., 2000).

- **Control of Animation.** Users should be able to create their own datasets when using the animation system. Since predefined data sets can also help users to build up initial understanding, the system should have both options.
- **Interactivity**. The animation system should provide forms of interaction between the user and the system. Interaction should include the ability to stop, undo, rewind or fast forward the animation to any desirable state.
- **History**. The animation system should enable users to return to and view previous states of the animation in execution.
- **Levels of Granularity**. Different levels are needed on both micro and macro scales so that the big picture can be understood from the bottom-up.

Morrison et al. (2000) presented a comprehensive investigation on the effectiveness of animated graphics in conveying visual or spatial information, stating that motion does not always automatically improve information transfer: similar to static representations, only carefully designed and appropriate graphics seem to perform better. In fact, animations often violate the second principle of good graphics, the *apprehension principle*, according to which graphics should be accurately perceived and appropriately conceived: animations are often too complex or too fast to be correctly perceived, and continuous events are sometimes conceived as sequences of discrete steps (Tversky, et al., 2002). Notably, the use of *interactivity* may overcome both these disadvantages, and the animation itself should always be implemented as a smooth and continuous process. Other experiments investigated the variety of roles for animation at the interface level (Baecker and Small, 1991).

Simple motions, such as linear oscillation about a point, are processed *pre-attentively* (Nakayama and Silverman, 1986). Consequently, motion can be used to support the understanding of overlapping graph diagrams (Bovey, et al., 2003). It has been shown that motion allows for *brushing* objects, an interaction process that enables the user to highlight, select or delete a subset of elements with a pointing device (Wills, 1996). Bartram (2001) conducted an exhaustive study on the use of motion for information visualization purposes, especially on its potential for filtering and brushing techniques. Her research proved that motion has a strong and effective *grouping* effect and converged in four guidelines for applying motion for filtering and brushing purposes (Bartram and Ware, 2002).

- **Similar motions are effective in perceptual grouping.** Things that move *together* in a similar fashion suggest to belong to a group.
- **Efficient motion grouping requires coherence between moving elements.** A certain *similarity* of frequency, refresh rate and change is needed to perceptually group separate items.
- **Motion type is an excellent discriminating feature.** The *type* of motion, such as linear, circular and expansive/contractive can be effectively used to group elements.
- **Direction can be used to differentiate linear motions under appropriate conditions.** Linear motions are most easily discriminated when they occur in different spatial quadrants and separated by at least 16°.

Although limited to two-dimensional motion rendered on normal displays, one can reasonably assume that these research findings have great value for all visualization techniques that employ motion features. Bartram also considered that the use of motion might encounter some disadvantages for visualizing informational values (instead of just brushing and filtering) as it would collide with the mapping of data onto *fixed* spatial positions: obviously, by moving away from the spatially mapped position, the object would loose its visual representation value. She did not consider, however, the use of motion for visualization techniques that do not require direct spatial mapping algorithms, probably because these are rarely used or explored. In effect, the infoticle metaphor will extensively use Bartram's guidelines for visualizing data values by differences in motion typology.

3.2.3. Behavior

Normally, animated objects follow certain defined paths that are based on various mathematical equations or constraints. In contrast, motion can also be controlled by so-called *behavior functions*, which tell objects how to behave based on the status of other objects in their environment rather than stating where to be exactly at specific times. The goal of this approach is to generate interpretatively rich and unexpected behaviors that seem to be *intentional*, by provoking the perception of causality, animacy and initiative.

Michotte (1963) suggests that *causal* relationships are perceived directly when certain simple animation techniques are used, such as launching, entraining and triggering. This phenomenon proves the potential of the rich expressive vocabulary of motion for information visualization purposes (Ware, et al., 1999). Lethbridge and Ware (1990) used simple behavior functions based on distance, velocity and direction to model complicated relationships such as pulling, pushing, chasing, escaping, repulsion, collision and anticipation. Interestingly, they describe a process of trial-and-error experimentation and pure chance as the main drivers of the design, as it seemed impossible to cast a particular behavior directly into a function. Emotionally, people divided movement into four different conceptual characteristics: the *path* that a moving object describes, the *area* that is covered during the movement, the *direction* of the animation and the *speed* and *tempo* of the object (Vaughan, 1997). For instance, erect, open and slow movement was considered as beautiful, while narrow, cramped and jerky motion provoked ugly and mechanic associations. In addition, this research concluded that complex patterns of movement tend to produce more connotative meanings.

Behavioral animation uses *local rules* to determine the dynamic motion of actors that visually simulate mental processes. The most common method of action simulation employs a *rule-based system*. Basically, a database is made up of a set of cause-and-effect rules to be followed by the objects that are being animated, usually called *actors*. In behavioral animation, an autonomous actor determines its own actions, at least to a certain extent. This gives the character some ability to improvise, and frees the animator from the need to specify each detail of every object's motion. For instance, *swarming* or *flocking* is a typical motion behavior resembling that of flocks of birds or swarms of fish and can be easily interpreted by humans. Accordingly, swarming can be characterized as *ordered* or *chaotic*, *tight* or *loose* and *global* or *regional*, for the goal of visualizing and analyzing network communication link establishment (Kadrovach and Lamont, 2002).

3.2.4. Conclusion

As mentioned before, both motion and stereoscopic vision are pre-attentive features. Information visualization extensively uses visual features that are pre-attentively processed, such as the position, length, size, orientation, contrast, curvature and hue of graphical objects. Consequently, one could easily assume that visualizing information by combining motion with traditional visual attributes and virtual reality display technology automatically will result in a pre-attentive and intuitive representation. However, pre-attentive processing in general only works for a single feature in well-considered cases and can easily be broken by simultaneous combinations of pre-attentive features that have proven to be pre-attentive individually, although a few interesting exceptions do exist. For instance, visual searches can be pre-attentive when combining spatially encoded information, such as two-dimensional positions, stereoscopic depth, or motion with a second attribute such as color or shape (Ware, 2000). In most other cases, humans cannot perform a conjunctive search effectively and require a focused serial attention span.

These research findings prove the potential of motion to represent dynamic data, as it is inherently able to give a good impression of data volumes, direction of flow and general, large-scale patterns of activity. Empirical studies show that motion is more effective than static graphics in conveying complex systems that involve real reorientations in space and time. In relation to the infoticle visualization technique, these studies suggest that users are cognitively able to perceive a large amount of points in stereoscopic environments, visually group similar trajectories, and recognize and interpret different characteristics of motions. Although movement is extremely powerful in gaining attention and communicating information, it can be easily abused as an uncomfortable factor. In short, the following design rationale rules should be considered.

- **Perception.** Stereoscopic vision increases the size of virtual space that can be understood, facilitates contextual overview, enhances motion perception and avoids spatial occlusions. Immersion enhances cluster recognition but makes effective interaction difficult. Motion paths that do not relate to mass and the object center can be easier understood. Complex motions result in richer interpretations. Pre-attentive processing can be broken by certain combinations of visual features. Motion without center-of-mass can be understood more effectively. Complex motion patterns produce more connotative meanings.

- **Data Discrimination.** Data can be effectively separated by motion typology, behavior typology, motion coherence, movement direction and spatial proximity. Motion interpretation is characterized by path, area, direction and speed (Figure 3.8). Motion can be compared, and visually grouped, by speed, acceleration, proximity and similarity in time.
- **Interaction.** Direct interaction with the simulation of time enhances the accuracy of motion, reduces the dynamic complexity and facilitates the understanding of short events and long trends.
- **Data Visualization.** The use of motion and spatial coordinate mapping algorithms might collide. Therefore, alternative visualization approaches are required to represent information values by motion typology instead of spatial locality.

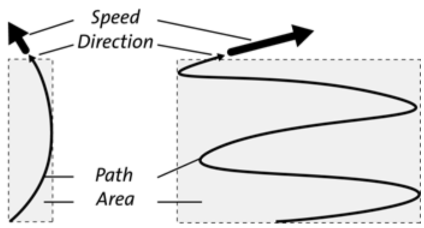


Figure 3.8. Motion behavior typology.

3.3. Concept

This section explains how the combination of particle animation, behavior generation and dynamic data mapping forms the core foundation of the infoticle visualization metaphor. The descriptions have been kept general to demonstrate the versatility of the proposed methodology. Although the infoticle system is essentially based on a set of rigid structural elements, it still has the intrinsic capability to be applied in a flexible way, depending on the dataset characteristics, application goals and the presentation medium. The infoticle framework is capable to alter the emerging visualization patterns or data interpretation algorithms by slightly adapting the metaphor behavior rules that control the attributes of every single infoticle.

3.3.1. Structure

An infoticle application is guided by three different principal processes. The *data process* is responsible for querying and caching the data from the remote database, while a separate *interaction process* evaluates all the events generated by the input devices. Programmatically as well as conceptually, the *infoticle application* process itself is composed of different hierarchical *classes* that enable an effective and smooth dynamic visualization simulation. Each single class defines specific data attributes and functions as a conceptual interface between all elements that make up an infoticle visualization system. From top to bottom, the hierarchical infoticle structure consists of the following elements (Figure 3.9).

- **Application.** This layer controls and halts the data and interaction process. It configures the technical attributes of the application, such as sound and visual settings, and stores the system-wide state variables. Practically, it is responsible for the top-level update loop that is executed once at each application frame update. In addition, it converts all the events generated by the input devices into suitable navigation and system reactions.
- **World.** The world level comprises all objects inside the virtual scene, such as particle emitters, tools, legends and so on. The world level can thus be imagined as the top of the infoticle system scene graph. It keeps track of the global variables, and continuously updates all elements that are present inside the virtual world. Therefore, each infoticle application consists of a single world class.
- **Emitter.** An emitter consists of a collection of infoticles, and consequently represents a specific dataset, which is typically stored in a single *database table*. This principle is described in Section 3.3.2.
- **Infoticle.** An infoticle is a dynamic particle representation of a time-varying *data object*, as explained in Section 3.3.3.
- **Tools.** Infoticle tools are interface and visualization elements that support users in comprehending the data patterns by spatially organizing the world layer. Tools include forces, filters, traces and shapes, and will be described in more detail in Section 3.4. They are placed within the scene itself, and are therefore members of the world.
- **Speaker.** A speaker can be imagined as a programmatic object that facilitates the communication with the parallel data process. In practice, it collects the data from the client and shares it with the infoticles of a single emitter. Normally, each emitter corresponds to a single speaker, which in turns corresponds to a specific *data attribute query*.
- **Client.** Each speaker in turn matches with a single client. All clients reside at the data process and handle the update and transfer of the database queries to the remote database machine. Each client awaits the data entries that come back, and parses them in a readable format to its corresponding speaker. In practice, the world class tracks whether all emitters have processed all the available data and then signals the data process to cache the data of the next timeframe. Alternatively, the data process can also check whether the data cache is empty, and start the query itself. Subsequently, each client sends a query to the database, after which the resulting dataset is converted and transferred to the shared memory on the local machine. The procedures involving the processing of data by speakers and clients are explained in detail in Section 3.7.3.

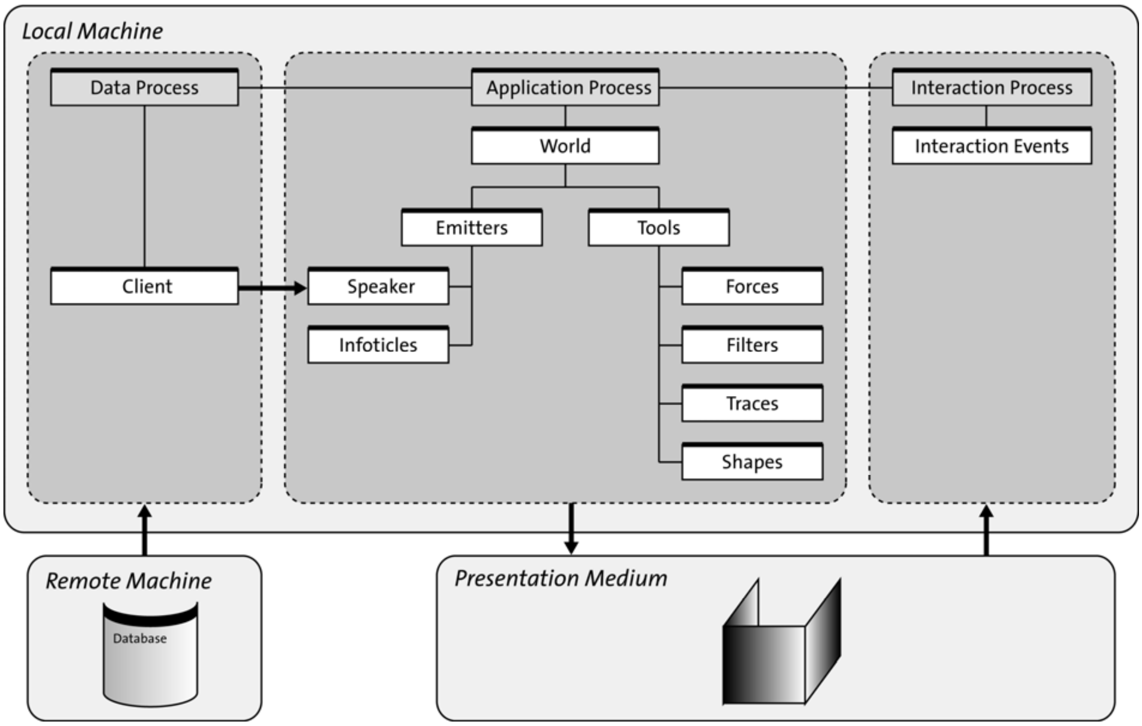


Figure 3.9. Infoticle structure.

3.3.2. System

The infoticle metaphor is capable of visualizing various time-varying datasets that are stored in remote databases. A *database* can be considered as a set of individual entries, along with a set of *links* between pairs of entries that indicate some particular relationship between them. A database can be easily and efficiently *queried*, resulting in subsets of entries that maintain the intrinsic knowledge of all links involved. The infoticle method uses databases instead of static data files for their versatile and flexible features. First, by automatically cross-matching meaningful connections between several related raw data files, datasets can be efficiently acquired and interpreted. Databases can be queried for subunits of data on the fly, so that different data attributes of the same dataset can be analyzed at various levels of detail. In addition, the calculation-intensive effort of browsing, searching, ordering and organizing large quantities of data is transferred to well-established, optimized database algorithms. The data within a database can be easily changed, extended or updated, even in real time, by a parallel, external process independent from the application. Furthermore, the start or the direction of the time simulation can be changed by adapting the query, so that data e.g. is retrieved in a reverse time direction. On the application level, a database is technically able to store subsets of data that have been selected by users during the visualization, which can be recovered afterwards by other media for further analysis.

An *infoticle emitter*, also called *infoticle system*, consists of a collection of infoticles and can be imagined as the visual representation of a single database *table*. This means that conceptually different datasets are not represented by a single emitter, but instead by several separate emitters. Much like traditional particle emitters, an infoticle system is composed of one or more individual infoticles that are influenced by three successive phases (Martin, 1999).

- **Generation.** Infoticles are generated at a predefined location within the scene. Each of the infoticle attributes is given default or random initial values. In practice, the system queries all available unique data objects in the database table and creates an infoticle for each single one of them.
- **Behavior.** During this stage, the attributes of an infoticle are determined by data object variables and behavior rules. The infoticle position is dependent on the data values, the previous state of the infoticle and the simulated time.
- **Dynamics.** This stage simulates the movement of an infoticle similarly to a traditional particle engine. The infoticle attributes are influenced by a time-varying evolution, other infoticles or external force elements, such as gravity, attracting or repulsing point forces and boundary surfaces. Therefore, the infoticle position is dependent on the previous particle position and velocity as well as time.
- **Extinction.** In some infoticle systems, an infoticle *dies* when its age matches a predefined lifetime value, which is the maximum time span that a particle can live (measured in frames or physical time units). Depending on the application, either the infoticle is removed from the scene and replaced by another infoticle representing the next timeframe, or its extinction might trigger the retrieval of new data values of the next timeframe for the same data object. Consequently, such systems consist of infoticles that never reach the extinction stage.

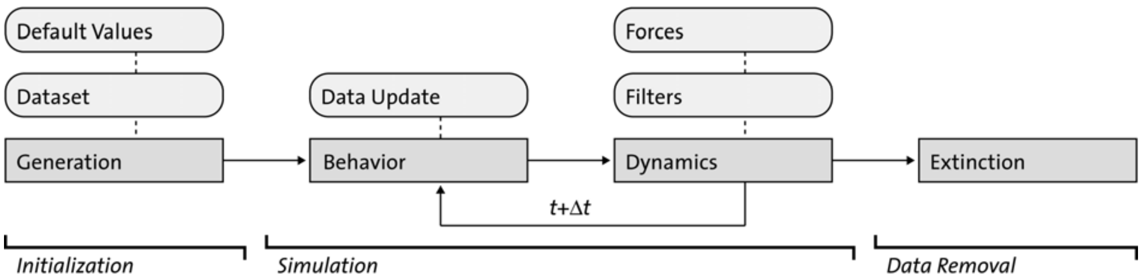


Figure 3.10. Infoticle life.

Figure 3.10 shows the previously mentioned infoticle life stages, and demonstrates the conceptual and programmatic distinctions between the parts that are similar to a typical particle engine simulation and those that determine the infoticle data mapping metaphor. In effect, the core of the infoticle data mapping method is mainly embedded in the behavior stage and is not fundamentally different from a traditional particle simulation.

3.3.3. Attributes

The infoticle simulation framework is derived from the traditional particle representation description. Consequently, an infoticle contains most of the normal particle attributes that are needed to effectively simulate a particle system, such as the current position, previous position, speed, direction, age, mass, lifespan, color, etc.

As Figure 3.11 demonstrates, infoticles can be visually rendered in different ways. An infoticle is typically represented by a small straight line that is blended into the background color. The *line representation* enables the simple recognition of the current direction, whereas its length denotes the actual speed. The *spline representation* connects the previously passed points in space. An infoticle can also be rendered as a half-transparent textured triangular polygon. This representation does not depict any directionality, but instead visualizes densities of spatial clusters by blending the colors of overlapping infoticles.

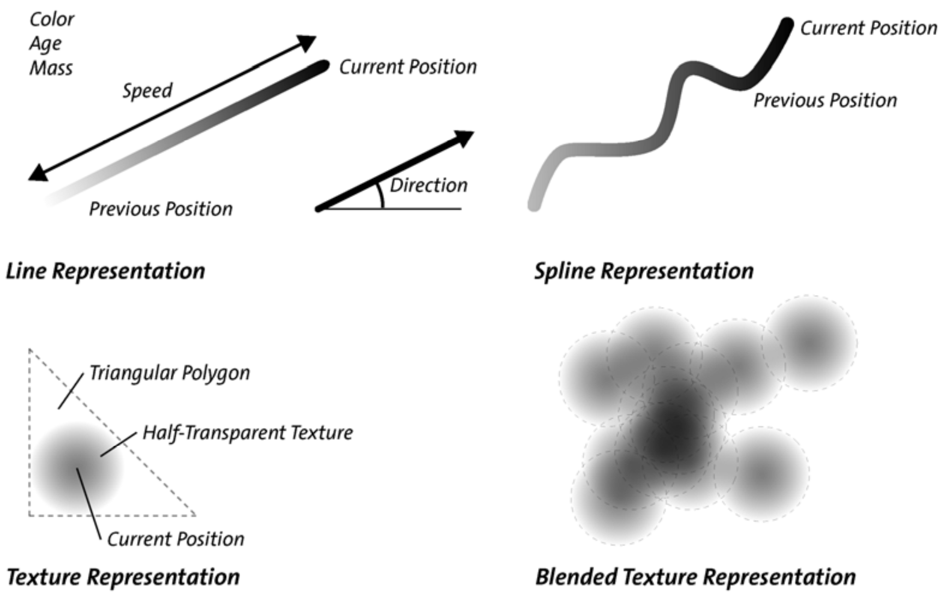


Figure 3.11. Infoticle representation.

An infoticle merges the concept of traditional particle systems with dynamic information processing. Each infoticle is a graphical representation of the time-varying data values of a specific, unique data object. In addition to the traditional particle variables, several specialized elements are required to effectively simulate an infoticle system. In practice, each infoticle contains information about its corresponding *data object*, keeps a *history list* of the spatial positions and data values it has processed, stores the coordinates of its corresponding *average force*, and manages several behavioral state dependent variables.

- **Data Object.** Each infoticle corresponds, conceptually as well as programmatically, to a single and unique *data object*. A typical data object consists of a fixed list of *data attributes* that contain specific *data values*. Practically, a data object resembles a row in a database table. Individual data objects are identified through a unique identifier. As shown in Figure 3.12, each data object points to a member of an object-oriented class that describes the attributes of the dataset within the database. During the visualization simulation, the data attributes of such a data object remain unchanged, but its exact data values alter in time due to the continuous time-ordered stream of newly updated information that is cached continuously from the database. In practice, one can imagine a data object to represent a single row of a database table. Such a database can either be static and thus contain the time-varying, historical data values within a certain historical time period, or can be updated by an external, real-time data source process.

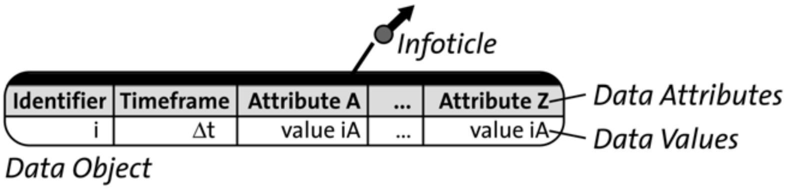


Figure 3.12. Data object.

- **History List.** A history list traces the corresponding infoticle back in time. It keeps track of the exact data values that have been evaluated and the spatial behavior that emerged out of this process. A history list ensures the reproducibility of the infoticle trajectory during the manipulation of the visualization simulation. It also enables the generation of detailed text information labels that show users the past data values that the data object was subjected to at specific points in space and time. A history list contains, ordered in time, historical data that resulted from the processing of the updated data values by the corresponding infoticle. This data is continuously accumulated and becomes updated at a predefined rhythm during the visualization simulation. A typical history list includes: the three-dimensional coordinates, the direction and speed values at the spatial positions the infoticle traversed, the averaged forces for multiple occurrences of data values and the specific data values it represented at these specific points in space and time (Figure 3.13). Section 4.5.3. Infoticle – Simulation – Time explains the time-related issues in more detail.

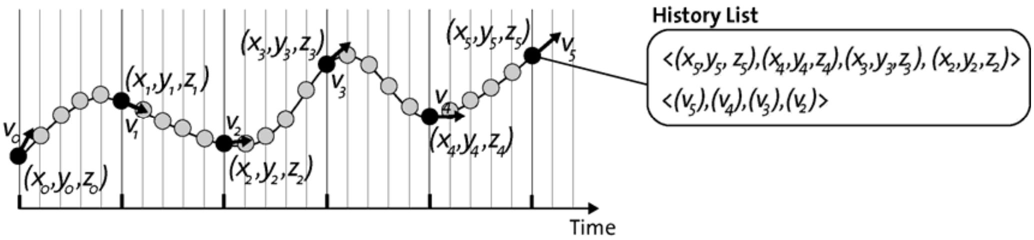


Figure 3.13. History List.

- **Average Force.** Infoticles are continuously influenced by forces with equal data values that are present within the scene. Consequently, once an infoticle becomes redirected by a different force due to a recent data update, it looses the visual representational value of the previous data from the moment it changes its spatial direction and leaves the vicinity of the formerly active force. However, it is meaningful to represent values that are valid at the current timeframe without loosing the context of the past historical data values. Therefore, the average force is unique for each infoticle and spatially represents the past data values the infoticle was exposed to. A detailed description of the average force concept will be given in Section 3.4.3. Infoticle – Tools – Force.

– **Behavior State & Application Variables.** In order to optimize the data update algorithm, some variables are needed that offer more information about the actual infoticle state. For instance, such variables track whether or not the data object of the infoticle has already been updated for the actual timeframe, so that the system does not need to repeatedly and unnecessarily invoke the search function for this specific data object. Other variables within this category offer information about the evolution of the actual infoticle behavior state, or store the reoccurring frequencies of a data object within single timeframes.

3.3.4. Initialization

During the first phase of the visualization simulation, all infoticles need to correspond to a single, identifiable data object. Depending on time-varying dataset characteristics, two sorts of infoticle initializations can be distinguished: *atomic* (one data entry matches one infoticle) or *multiplied* (one data entry matches many infoticles).

3.3.4.1. Atomic

The atomic infoticle mapping mechanism is characterized by a *singular* relationship between the database and an infoticle. Figure 3.14 shows that during the initial generation process, each database object, or in practice each database row, within the first database timeframe corresponds to a single infoticle. This singular infoticle versus data object mapping mechanism is suited to visualize individual entities of abstract data that are unique and cannot be divided nor grouped by specific quantifiable units, such as persons, documents, network packets and so on.

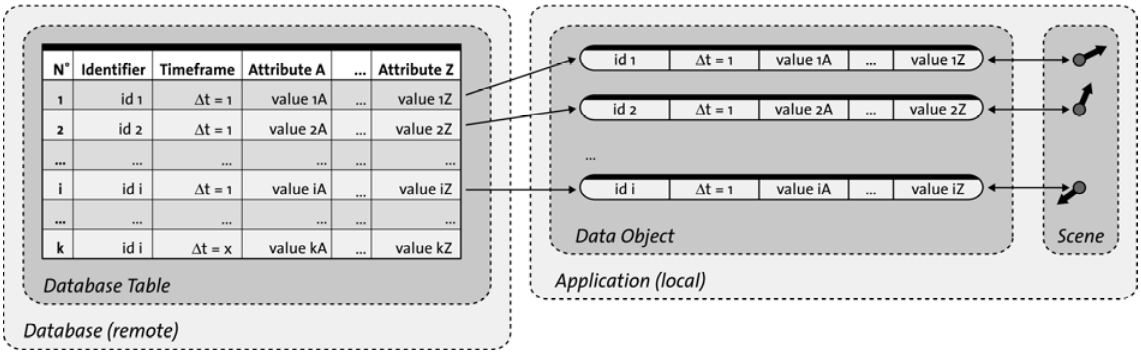


Figure 3.14. Atomic infoticle initialization.

3.3.4.2. Multiplied

The previous data mapping method can be extended to encompass the visual representation of quantitative data. This approach results in a *one-to-many* relationship between a single database entry and a set of corresponding infoticles. In practice, this means that a specific quantifiable data value within a database row is divided into a discrete amount of corresponding infoticles. This mapping technique facilitates the visualization of data values that cannot be clearly divided into smallest-order entities for various reasons. For instance, the quantity of 100 persons can theoretically be separated into 100 unique data objects that are each identifiable by their name and address, although this information might not be available because of certain privacy concerns or imprecise tracking tools.

Notably, these datasets are represented in traditional visualizations by object size, graph height or other measurable graphical cues that enable the visual comparison of numerical values. In contrast, Figure 3.15 demonstrates the mapping of a single quantifiable data value onto a corresponding number of infoticles, representing atomic data objects that are equal at initialization, but can be altered individually later on.

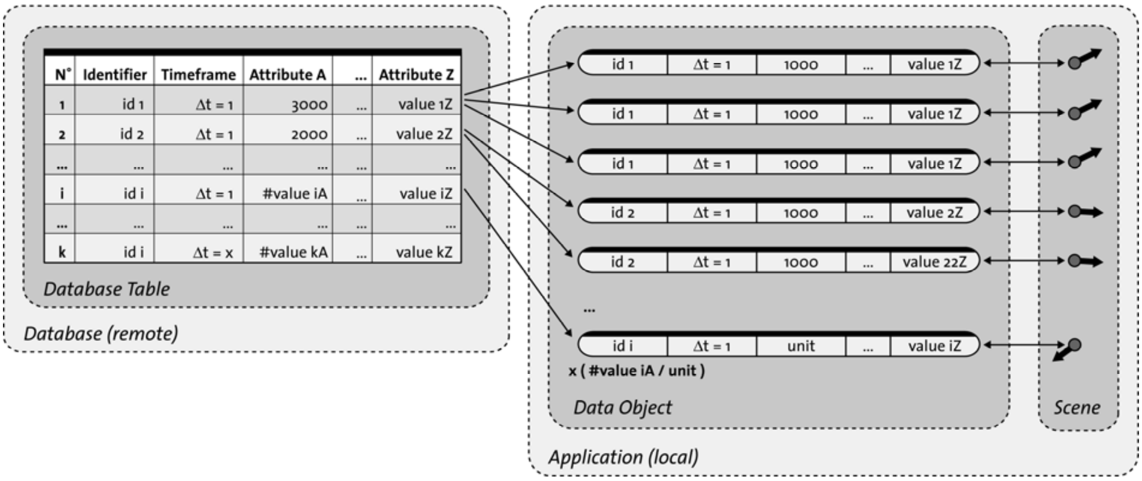


Figure 3.15. Multiplied infoticle initialization.

By directly translating numerical values into exact quantities of individual elements, this visualization method facilitates a direct and intuitive visual comparison of large data entry streams. Users are able to observe changes in density and cluster volumes, and can easily influence the many infoticles within the spatial flows. This mapping mechanism is not suited to directly measure the *exact* quantities of infoticles within certain constellations but instead facilitates the immediate and measurable perception of relatively small changes in data, for instance when a small fraction of infoticles suddenly behaves differently.

3.4. Tools

An effective information visualization application needs various features that allow for direct manipulation of the data mapping metaphor to explore both the logic of the method and the resulting representation space. Within the context of this research, *tools* are those elements that influence spatial infoticle trajectories and support visual pattern interpretation. In practice, they make up the major part of the infoticle method interface. Such features should neither break the metaphor consistency, nor disturb the intuitive interaction with the representation. Therefore, the infoticle metaphor attempts to maintain the feeling of presence by presenting the tools within the three-dimensional scene.

3.4.1. Source

At the initialization of the visualization world, infoticles emerge in the virtual world at the *infoticle emitter source* position. New infoticles are created with a certain frequency, which determines the amount of new particles per frame. Infoticles stream into the world with initial spatial direction and speed values randomized between predefined thresholds, preventing the creation of multiple infoticles that possess equal spatial positions. As Figure 3.16 illustrates, the spatial directions lay always inside a *particle expulsion cone* that is determined by specific *pitch* and *yaw variation* values. Typically, infoticles stream out an emitter in a time-ordered way.

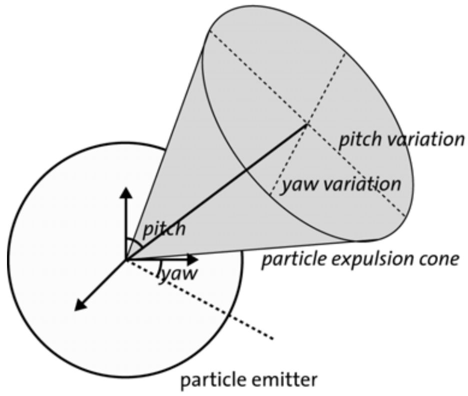


Figure 3.16. Infoticle source.

The emitter source position plays a minor role in those virtual worlds that simulate infoticles with an infinite lifespan. In such visualization scenarios, no new infoticles are dynamically added or removed, so that the initial creation position has no noticeable repercussions on emerging visual patterns. Some infoticle visualization simulations however require infoticles to be created and recreated at predefined points in time, so that users can observe the outcome of specific sequences and adapt the spatial tool constellations to their own data exploration expectations. Such scenarios require the direct manipulation of both the position and direction of all infoticle emitter sources.

3.4.2. Filter

One of the most important user tasks in information visualization applications is called *data filtering*. Filtering is an interactive process that typically reduces the amount of context in the display by decreasing the number of data points. It is employed when the dataset is too large or too complex to display, or when the user is interested in a certain subset of the data, or is unable to effectively comprehend the information due to visual occlusion or cognitive data overload.

Visualization applications generally use sliders or buttons for defining the specific attribute range to compare the dataset with, so that those data points that do not fulfill the filtering rule disappear from view. In contrast, the infoticle metaphor represents data filters as flat rectangular surfaces that are located within the three-dimensional scene, avoiding the need to display separate interface tools such as text menus or widgets.

Conceptually, each filter embodies a typical conditional *if-then-else-clause*, which corresponds to a set of data attributes of each infoticle. A filter rule can consist of either a single atomic value in case of a nominal dataset, or a specific mathematical range in case of a quantitative dataset. One can imagine an infoticle filter as a human-sized door, which lets through all data objects that comply with the filter condition and physically bounces back all data objects that are different. In practice, such filters act like virtual borders that divide the space into zones of equal data values. Figure 3.17 shows how the visual *collision* or *bounce* metaphor is employed to filter abstract data values within a three-dimensional information environment.

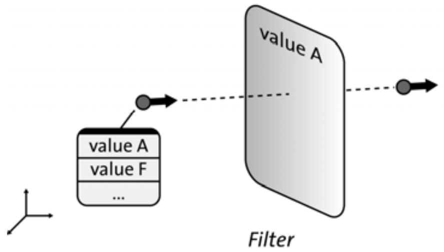


Figure 3.17. Filter concept.

This filtering metaphor appeals to the intuitive notion of defining surfaces that border private spaces in physical reality. By manipulating the spatial position and orientation of the filters, users are able to model a personalized information environment that spatially categorizes infoticles by their data values. The three-dimensional representation of the filtering surfaces provokes users to carefully consider the spatial design of the elements that are present within the virtual world.

Instead of creating abstract Boolean rules and data queries using text interfaces or two-dimensional graphs, users are instead encouraged to *spatially model* an environment that effectively redirects, groups and filters data objects. Figure 3.18 shows how by placing filters in well-considered constellations, infoticles with different data values can be directed towards the outskirts of the visualization or become merged into specific regions of interest. Similarly, specific spatial arrangements can be related to traditional data queries. For instance, positioning several filters with increasingly detailed data values in a successive row automatically generates a spatially induced information content refinement.

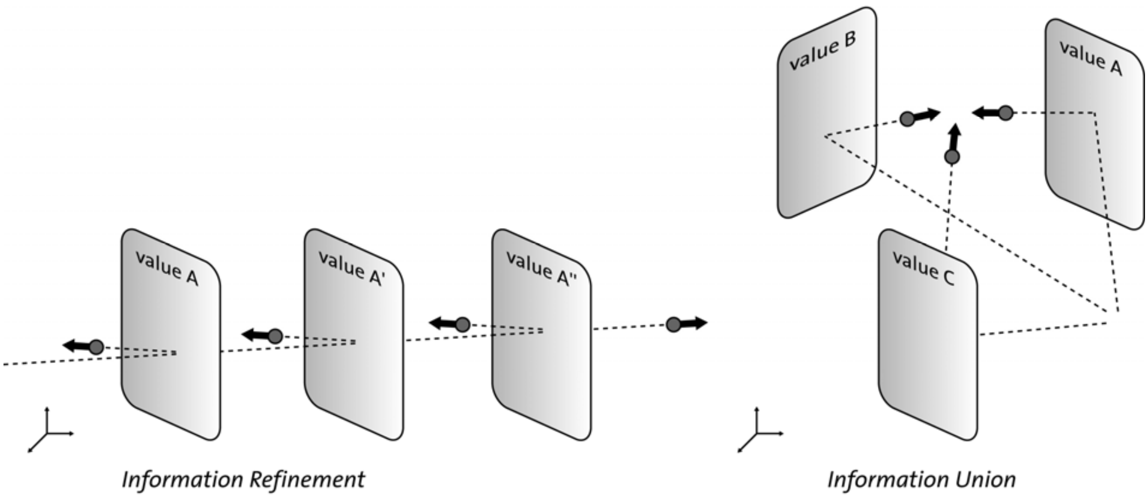


Figure 3.18. Filter constellations.

The infoticle filter method merges abstract thinking with spatial designing, thereby using the human cognitive capabilities of space perception and space modeling. In effect, the resulting virtual space is not determined by aesthetic contemplations or traditional physical limitations such as gravitation and construction stability. Instead, an infoticle world design complies with the rules of effective information organization and presentation. It can be directly adapted by users according to their personal data exploration goals.

Unlike most visualization applications, the filtering rule is not checked for the whole data collection instantaneously, but only for those infoticles that have *just* collided with the three-dimensional surface plane. This process requires real-time *collision detection* and data value comparisons instead of traditional, globally valid, pre-calculated filtering dependencies. It enables the filtering of dynamic or real-time datasets, as newly arrived data values are continuously influenced by all filters within the scene. However, this method also results in a calculation-intensive computing process, as the system needs to check every new infoticle position for a possible filter collision at each single frame.

Figure 3.19 shows the typical problem of collision detection in computer-generated animations. As in computer graphics, time is tested in discrete intervals, a possible collision might be missed if the according procedure is not properly implemented.

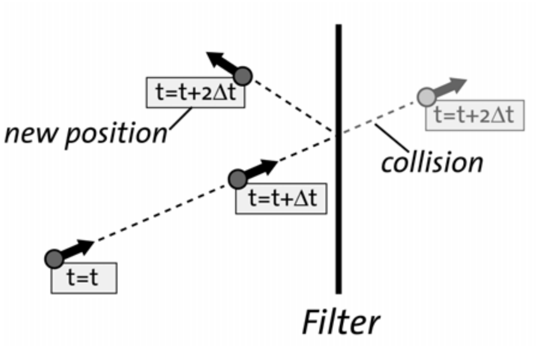


Figure 3.19. Time-discrete filter collision detection.

A three-dimensional plane can be mathematically described by the direction of its normal $\vec{N}_{plane}$ and the distance D_{plane} from the plane origin $\vec{P}_{plane}$ to the world origin. The point distance *d* to a plane from a point $\vec{p}$ outside the plane can be found by taking the dot product of the plane's normal vector $\vec{N}_{plane}$ and subtracting the distance D_{plane} of the plane to the origin.

$$\vec{N}_{plane} \cdot \vec{P}_{plane} - D_{plane} = 0$$

$$d_i = \vec{N}_{plane} \cdot \vec{p}_i - D_{plane}$$

In practice, the previous and newly calculated positions of an infoticle are checked against the mathematical description of the three-dimensional surface direction. If an infoticle passes through this infinite plane, the combination of this formula with the two successive positions results in opposite mathematical signs.

$$\left.\begin{aligned} d_{previous} &= \vec{p}_{previous} \cdot \vec{N}_{plane} + D_{plane} \\ d_{current} &= \vec{p}_{current} \cdot \vec{N}_{plane} + D_{plane} \end{aligned}\right\} \Rightarrow d_{previous} * d_{current} < 0\,?$$

If this is true, the exact collision position is calculated by measuring the relative distances to the force from the previous and next infoticle position.

$$\left.\begin{aligned} t_1 &= \vec{N}_{plane} \cdot (\vec{P}_{plane} - \vec{p}_{previous}) \\ t_2 &= \vec{N}_{plane} \cdot (\vec{P}_{plane} - \vec{p}_{current}) \end{aligned}\right\} \Rightarrow \vec{p}_{hit} = \vec{p}_{previous} + \left(\frac{t_1}{t_2}\right) \cdot \left(\vec{p}_{current} - \vec{p}_{previous}\right)$$

Subsequently, it is tested whether the infoticle passed within the spatial borders of the surface. First, the offset distance $\vec{d}_{offset}$ between the plane origin and the hitpoint is calculated, and expressed in terms of the plane's basis vectors. When the corresponding hitpoint coordinates lay within the size of the plane, and thus are smaller than 1, the infoticle crossed the spatial borders of the plane.

$$\left.\begin{aligned} \vec{d}_{offset} &= \vec{p}_{hit} - \vec{P}_{plane} \\ x_{hit} &= \vec{d}_{offset} \cdot \vec{u}_{filter} \\ y_{hit} &= \vec{d}_{offset} \cdot \vec{v}_{filter} \end{aligned}\right\} \Rightarrow \left.\begin{matrix} 0 < x_{hit} < 1 \\ and \\ 0 < y_{hit} < 1 \end{matrix}\right\} \Rightarrow hit!$$

Only then will the system compare the data values of the infoticle with the corresponding conditional filtering rule. When this comparison returns positive, the bounce vector is calculated. First, the bounce normal component $\vec{n}_{hit}$ and the tangent component $\vec{t}_{hit}$ are calculated out of the magnitude m_{hit} of the bouncing vector in the direction of the plane normal $\vec{N}_{plane}$. Then, the new infoticle direction $\vec{v}_{new}$ equals to the tangent and normal components of the bounce vector direction, damped by the damping coefficient $\omega_{damping}$, which is added to the exact collision coordinates on the surface, resulting in the corrected position coordinates $\vec{p}_{new}$.

$$\begin{aligned} m_{hit} &= \vec{v}_{current} \cdot \vec{N}_{plane} \\ \vec{n}_{hit} &= \vec{N}_{plane} . m_{hit} \\ \vec{t}_{hit} &= \vec{v}_{current} - \vec{n}_{hit} \\ &\Downarrow \\ \vec{v}_{new} &= \vec{t}_{hit} - \vec{n}_{hit} . \omega_{damping} \\ \vec{p}_{new} &= \vec{p}_{hit} + \vec{v}_{new} \end{aligned}$$

Experience has shown that it is often more effective to switch the order of the previous algorithms, i.e. to invoke the data-filtering rule first, which consists of a simple numerical or textual comparison, and subsequently the collision detection, which requires more complex mathematical formulas.

3.4.3. Force

Clustering or *perceptual grouping* is another important visualization method to represent similarity between data points. This technique uses the Gestalt rule of *common fate* that describes the grouping of elements with similar form, color, brightness, size, speed, motion typology or spatial orientation. In addition, the Gestalt law of *proximity* states the automatic visual clustering of elements that are positioned at short distance to one another.

3.4.3.1. Point

The infoticle metaphor uses the force concept to redirect and group infoticles with similar data values. A *force* can be imagined as an external, mathematical point in space that attracts or repulses the infoticles within the scene according to the laws of Newtonian mechanics. Similarly to the data filter metaphor, each infoticle force embodies a specific if-then-else condition. As a result, only those infoticles that comply with this conditional clause are attracted to the force, all others remain unaffected (see Figure 3.20). Such condition can either consist of a nominal data value, or contain a specific numerical range that is valid for a set of multiple data attributes.

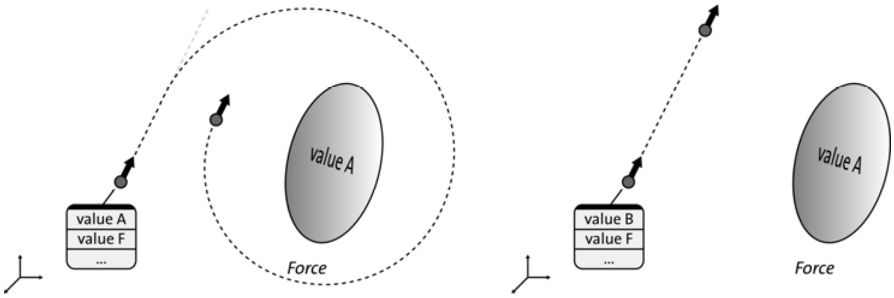


Figure 3.20. Force concept.

According to Newton's *Law of Motion*, a force is related to an object mass multiplied by its acceleration, which itself is the change in velocity. Consequently, given the mass m of an object and the attraction or repulsion strength vector $\vec{F}$ of the force, the acceleration a can be calculated and added to the current object velocity v, representing speed and direction, to result in the new velocity value. This velocity then determines the new spatial position $\vec{p}_{current}$ of the infoticle. As psychophysics experiments of dynamic behavior have shown that differences in mass result in subtle visual patterns that are difficult to perceive, the mass m_p of each infoticle and m_{force} of each force is considered to be constant and equal to 1.

$$\left.\begin{aligned}\vec{F} &= \frac{F_{strength}.m_{force}.m_p}{\left\|\vec{p}_{force} - \vec{p}_{previous}\right\|^2}\\ \vec{F} &= m_p.\vec{a} \Rightarrow \vec{a} = \frac{\vec{F}}{m_p}\end{aligned}\right\} \Rightarrow \vec{a} = \frac{F_{strength}}{\left\|\vec{p}_{force} - \vec{p}_{previous}\right\|^2} \quad \text{and} \quad \begin{aligned}\vec{v}_{current} &= \vec{v}_{previous} + \vec{a}.\Delta t\\ \vec{p}_{current} &= \vec{p}_{previous} + \vec{v}_{current}.\Delta t\end{aligned}$$

The use of point forces enables the clustering of infoticles in a smooth, animated way, clearly visualizing the different starting and ending states within a specific timeframe. Furthermore, this method overcomes many of the traditional force-directed visualization problems. In most force-directed visualization systems, spring stiffnesses correspond to similarity measures that need a certain dedicated pre-calculation time. An energy minimum is calculated out of these various spring relationships that reveals multi-dimensional relations and adjacencies in terms of spatial neighborhoods. In contrast, the infoticle force metaphor is unaware of any data similarities beforehand, as all infoticles behave independently of each other and the different force influences are simply added mathematically. In addition, the force simulation does not need a dedicated calculation time to create the scene and never reaches a static state of equilibrium.

It should be noted that, because Newtonian mechanics influences infoticle trajectories incrementally and smoothly, the resulting representation requires a certain time span until it reaches a *true state*. As will be demonstrated in Chapter 5, this adaptation time results in possibly wrong observations but also shows a powerful intrinsic potential for the clustering of similar long-term data evolutions.

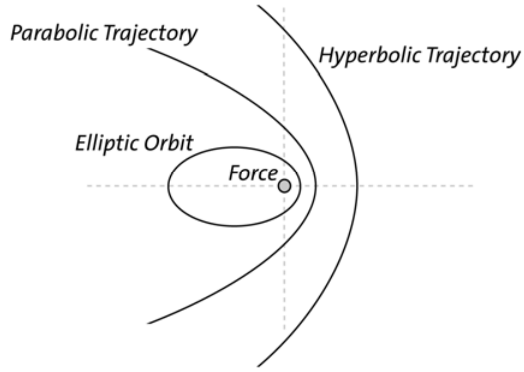


Figure 3.21. Force-influenced infoticle trajectories.

Figure 3.21 demonstrates how, depending on the initial speed, size and distance, the resulting trajectory of a particle influenced by a force results in a circular, elliptic, parabolic or hyperbolic trajectory. As the latter two behaviors catapult the infoticle away from the force hereby loosing the visual connotation, either the force parameters or infoticle behavior rules have to be adapted in such a way that infoticles cluster into circular or elliptical trajectories. These precautions not only result in clustering patterns that are easier to comprehend, but also determine the appropriate size of the representation, so that infoticles are always at a relatively short distance from the other forces when these should start to influence them as well.

Other measures are needed to avoid the occurrence of phenomena of unwished visual behavior. For instance, infoticles coming too close to a force might crash into the center, resulting in sudden infinite speed accelerations that would project infoticles to the far outskirts of the scene. Although these necessary adaptations might seem to break the simplicity of the overall point force metaphor, they only take place at an algorithmic level that is completely hidden from users. In practice, these dynamic behavior adaptations still result in a highly consistent data representation with easily comprehensible and identifiable visual patterns.

3.4.3.2. Time-Average

The average force is one of the core infoticle attributes. It is a time-averaged position in space that represents all forces the infoticle has been and currently is attracted to, as the actual position and direction of an infoticle only represents the momentarily valid data values of the active timeframe, because it spatially leaves those points it was attracted to before. When a new data object arrives, the time-average force $\vec{F}_{time}$ of a set of historical forces $\vec{F}_{average}$ (all data values averaged over a single timeframe and calculated in Section 3.5.2.) of an infoticle is calculated as follows.

$$\vec{F}_{time} = \frac{\sum_{i}^{n} \omega_i . \vec{F}_{average}}{\sum_{i}^{n} \omega_i}$$

n is the total number of historical average forces available within the infoticle history list, and the time-averaged weight ω_i is defined as:

$$\omega_i = \frac{A - \Delta t}{A}, \omega_i > 0$$

A is a predefined time constant. For instance, if we assume that Δt is calculated in hours and *A* equals 24, then all data value alterations older than one day are not considered in the calculation of the average force.

3.4.4. Trace

Lines are a very powerful and common way of visualizing a conceptual connection between two or more related items. When used in the context of time, lines are able to act as spatial representations of dynamic change. Accordingly, altering data dimensions can be mapped onto the spatial directionality of lines, resulting in the concept of *timelines*. According to Maeda (2000), a digital stroke that is generated out of space and time considerations is not just a mark left in space, but a sculpture with interpretable space-time qualities.

As both scientific (see Section 2.4.3. Background – Particle – Scientific) and infoticle visualizations deal with small point objects that continuously move in three-dimensional space, some conceptual similarity exists in the usage of visual artifacts that represent spatial trajectories in a comprehensible way. The most common visualization method of motion tracing generates spatial *curves* or *splines* that sequentially connect all points in space that the infoticles have traversed. The visualization of such curves is a widely used technique in the area of scientific visualization, as these line representations are able to provide information about the character of unsteady, time-dependent flows. There are several different types of curves, depending on time characteristics and control points.

- **Pathlines** trace a particle released into a flow field over a period of time, assuming multiple time instants. In experimental visualization, this can also be achieved by long-term film exposure.
- **Streaklines** pass through all particles that are continuously inserted in a flow from one or more fixed positions over a period of time.
- **Timelines** connect the positions at an instant of time of a batch of particles that has been released simultaneously.
- **Streamlines** are integral curves of an instantaneous velocity field passing through a given point in space at a given point in time.

3.4.4.1. Line

The infoticle metaphor uses the concept of *pathlines*, as they provide the most suitable characteristics for visualizing the evolution in space and time. Pathlines also enable a simple visual comparison of individual infoticle trajectories instead of infoticle groups or batches that have some time-based relationship with one another.

Different curve varieties exist, depending on the function used to approximate the curve along the points defining the spline. Technically, an infoticle trace is calculated out of a so-called *Catmull-Rom spline*, which denotes the mathematical representation of a curve that connects a series of points at intervals along the curve. The Catmull-Rom technique was chosen because it generates smooth curves that pass through all defining points, are continuous in the tangent direction and have second-degree curvatures that change linearly over the length, as shown in Figure 3.22. Furthermore, Catmull-Rom nodes are visually attractive because they generate stable lines, even when the controlling points are in close to one another.

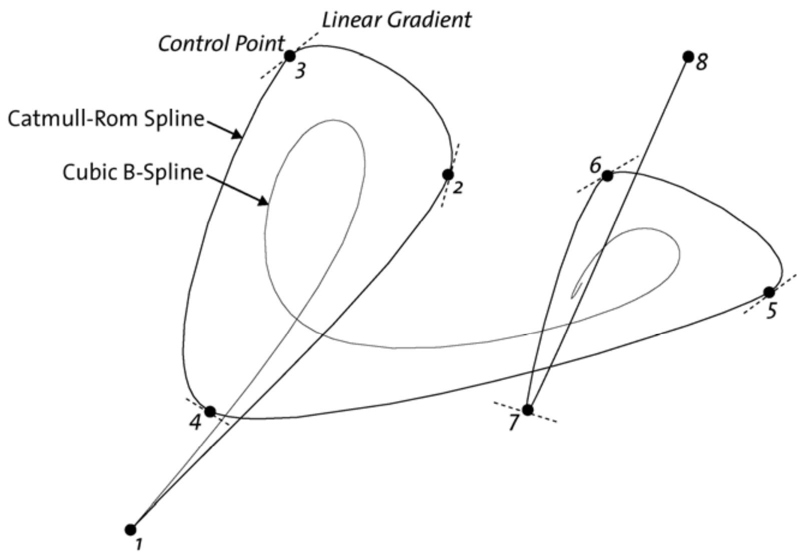


Figure 3.22. Catmull-Rom Spline.

Pathlines pass exactly through those points in space that the infoticle has reached before. In practice, the infoticle pathline uses the coordinates stored within the history list as control points. As a result, a smooth pathline is calculated that is a close approximation, but not an exact replica, of the path the infoticle followed. Figure 3.23 demonstrates this procedure, and shows how the resulting pathline spline slightly differs from the precise trajectory during the update simulation.

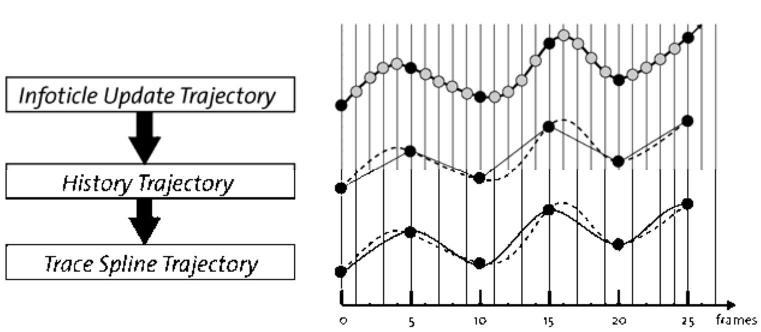


Figure 3.23. Pathline initialization.

3.4.4.2. Ribbon

In case of the infoticle method, the use of *ribbons* was preferred to that of lines, as they convey more spatial qualities and facilitate the representation of an extra data dimension. In scientific visualizations, such ribbons are also called *stream surfaces* or *stream polygons*. Stream surfaces connect various streamlines (Hultquist, 1992), whereas stream polygons are generated by sweeping a polygon along the streamline, varying the radius and shading the resulting surface (Schroeder, et al., 1991). In effect, ribbons are more easily understandable than lines, as their width component results in clearly perceivable spatial occlusions within three-dimensional space. An infoticle ribbon is typically constructed by connecting two Catmull-Rom splines parallel to the infoticle history pathline. An infoticle trace ribbon consists of the following visual attributes (Figure 3.24).

- **Width.** The dimension of the ribbon width can be altered along its path in a continuous and smooth way. The exact width size represents a specific quantifiable and time-varying data value of a data object attribute, defined by the application designer. Often, the ribbon width represents the frequency of the data object within the database frame for parallel sequential datasets (see Section 3.5.2.).
- **Color.** The color is equal to the according infoticle color. It denotes the particle emitter and thus the original dataset typology.
- **Transparency.** A decreasing transparency along the length of the ribbon pathway from start to end represents the direction of the passing time and the infoticle history.
- **Shape.** When considered in the context of time and space, the similarities between different ribbons can be compared. In addition, the overall shape typology represents the different motion characteristics an infoticle was subjected to and thus contains various perceivable informational values.
- **Labels.** Text labels along the path of the infoticle trace denote the three-dimensional points in space through which the infoticle has traversed. Additionally, they represent detailed textual information of the historical data values at those specific points in space and time.

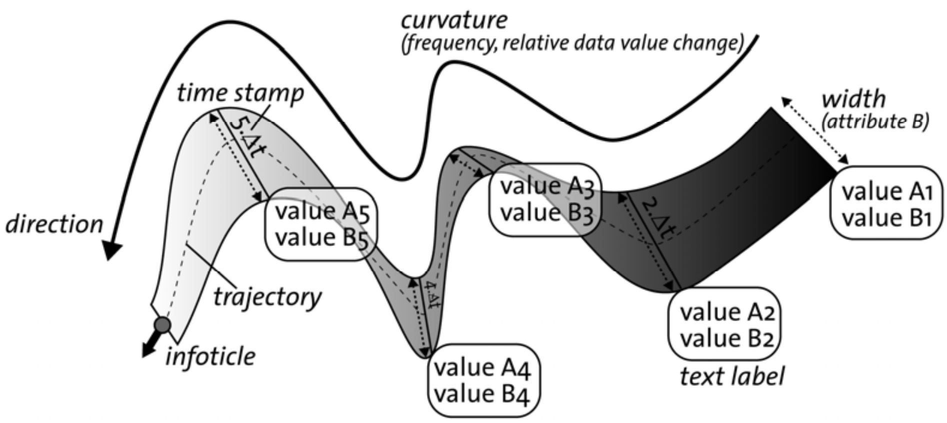


Figure 3.24. Ribbon visual attributes.

As will be demonstrated in Chapter 4, an infoticle trace shape can be interpreted in the context of various time-varying data update characteristics. Therefore, infoticle traces act as static, visual artifacts representing historical motion typologies. In effect, several different ribbon shape patterns can be cognitively recognized and linked to different meaningful types of dynamic dataset alterations.

3.4.5. Shape

The formal language of enclosed figures like blobs, circles, bars, lines, crosses and arrows, suggests certain physical properties that cognitively provoke compelling conceptual interpretations. Furthermore, enclosed figures suggest the possibility of *containing* certain elements, thus effectively separating those elements from others. All humans tend to impose imaginary forms, for instance on the irregularities of rocks, inkblots or clouds. This suggestive tendency, driven by an unconscious wish to organize the surrounding environment, can be so strong that random shapes trigger the perception of surprising illusions. By consequence, specific graphics designed with this phenomenon in mind might be used for effective virtual reality visualizations. Accordingly, the infoticle metaphor uses *blob shapes* to denote data similarity and to support the perception of spatial clusters.

3.4.5.1. Blob

A typical *blob* object is defined with a center, a surface area, a mass relative to other objects, and a specific field of influence. The field of influence defines a relational zone within which the blob will *fuse* with, or be inflected by, other blobs. When two or more blob objects are proximate they will either mutually redefine their respective surfaces based on particular gravitational properties, or actually fuse into one continuous surface defined by the interactions between their respective centers and zones of inflection and fusion. Blobs are often described as quasi-solid, and can only be understood in relation to other objects, as their center, surface area and mass are solely determined by various other elements of influence.

By enclosing infoticles within shapes, the visual perception of global and directional trends becomes more effective. In effect, a user is able to recognize the visual impact of a single shape more rapidly than the spatial clusters of small-sized infoticles. In addition, shapes reduce the amount of points to be observed, and shift the attention from individual movements to a more formal interpretation. These shapes also facilitate the detection of individual outlying infoticles that might otherwise have been overlooked: points that lay outside of the cluster cores typically form *bumps* or *bulges* that are relatively large and visually more predominant.

The infoticle metaphor does not touch upon some specialized aspects of multiple encapsulated shape generation for information visualization (Sprenger, 2002). For instance, rules of good visualization mention that (1) objects must belong to one shape only and multiple assignments are prohibited, and (2) shapes must not intersect. In fact, the infoticle metaphor avoids most of these issues by generating just a single, global shape at all times.

In addition, the infoticle method employs shape polygonizing algorithms that were designed for static surface generation, so that the calculation of time-varying surfaces requires a dedicated iterative sequence of computations. Also, one should note that accurate shape perception is considerably influenced by many external factors, such as orientation and lighting, and requires the user to observe the shapes from several viewing angles for a detailed formal interpretation.

3.4.5.2. Marching Cubes

The generation of infoticle blob shapes is based upon a specific surface mesh algorithm called *marching cubes*. This method is able to generate closed, three-dimensional surfaces that span a number of three-dimensional points (Lorensen and Cline, 1987), in this case the collection of infoticles. First, a mathematical iso-surface is generated that covers all the points in the scene. Then, this algorithm divides the three-dimensional space into a series of equal small cubes, and uses a divide-and-conquer approach to locate the possible surface in each of these cubes. Subsequently, a specific procedure *marches* through each of the cubes and tests whether the corner points of the cube lay inside or outside of the surface, or whether it intersects them. Since there are eight vertices in each cube and two possible states, inside and outside, there are only $2^8 = 256$ ways for a surface to intersect a cube. By enumerating these 256 cases, the possible surface-edge intersections can be compared with a predefined look-up table. By combining the solutions originating from this table of possible edge intersections, a global, polygonal structure is created which approximates the iso-surface. Several variables determine the exact size of these cubes, the amount of mesh coordinates and how close the surface needs to approximate the point constellation. The infoticle application is based upon the work on and the source code of an improved marching cubes method called *implicit surface polygonizer*, which defines the divisions dynamically using algorithms instead of a look-up table (Bloomenthal, 1988, Bloomenthal, 1994).

3.5. Simulation

The infoticle metaphor is explicitly designed to visualize the dynamic characteristics of time-varying datasets. A typical database table representing a time-varying dataset consists of multiple entries of a specific data object that is identified by a unique identifier. Consequently, these reoccurring data entries correspond to the changes in the data values for that specific data object in time, and the visualization metaphor should be able to accurately represent these dynamic occurrences.

This section describes the conceptual issues involved when visually simulating data evolutions in an incremental way. First, time-ordered data subsets need to be retrieved from the remote database and cached at the local visualization application. Then, the continuous data updates have to be visualized by interpreting the individual relative data value changes in real time. In case of the proposed infoticle metaphor, this data mapping is accomplished either by data-dependent behavior rules or by more subtle, internal relationships generating self-organizing, emergent patterns.

3.5.1. Data Flow

Each infoticle data object is subjected to a continuous stream of updated data values that correspond to the actual application timeline. The goal of the visualization metaphor is to represent the characteristics of this update process through an automatic interpretation process. Consequently, a conceptual flow from the original data source to a resulting representation has to be followed by all available data values.

The infoticle visualization method consists of four sequential layers that each data object has to traverse. These different stages, illustrated in Figure 3.25, demonstrate how the system is able to automatically translate raw data from an arbitrary source into comprehensive spatial behaviors that in turn generate an emergent, global representation. In addition, these layer descriptions clarify the conceptual implementation adaptations that are required depending on specific dataset characteristics.

- The **Acquisition Layer** includes all conversion actions needed to port the original raw data to the database. Typically, the acquired data is described by different protocols, stored within exotic file types and includes various invalid data entries. A process of filtering, cross-matching and type-converting is needed to translate the dataset in a standardized format that is readable by the infoticle data process algorithms. In general, each separate dataset is transformed into a single database table, which includes all the individual data object reoccurrences in time. The actions within this layer are typically accomplished well before the visualization. However, a real-time visualization of live datasets requires this stage to be processed in sequence with the subsequent layers.
- The **Collection Layer** retrieves subsets of data within specific database timeframes from a remote database, using dynamic, real-time data queries. In the case of time-varying datasets, this layer should be aware of the current system time state, so that new data can be cached in an efficient manner. Because of the large and unpredictable data amounts to be processed and the dependence on the network communication performance and remote machine calculation power that is running the actual database, this layer is executed completely separately and runs parallel to the other layers. This computing partition ensures a continuous and smooth data representation, even when the database connection is interrupted, the transferred data volume is huge or the data queries require time-intensive calculations on the database side.
- The **Structure Layer** consists of predefined rules that dynamically structure the incoming dataset into certain spatial behaviors of the corresponding infoticles. These principles determine how a changing data value must be interpreted, which infoticle attribute it will influence and what algorithmic effect it will have on that attribute. In practice, this layer completely controls the data visualization outcome. The structure layer has to be redesigned for every application, depending on the dataset characteristics and the data patterns expected to be hidden within. Application designers are required to alter the structure layer procedures to experiment with the effects of the behavior rules.
- The **Presentation Layer** is responsible for simulating and representing the visualization world. It includes the normal animation algorithms that apply when the data is not updated, and deals with typical application issues such as user interaction and navigation. Normally, only small alterations of this layer are required for each application, depending on dataset dimensionality, three-dimensional graphical layout or desired visual cues. This layer also includes the default spatial configuration of the tools within the scene.

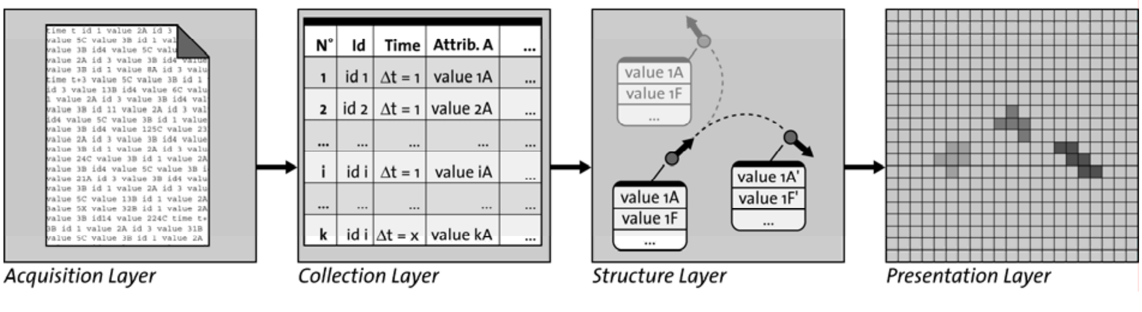


Figure 3.25. Infoticle data flow.

Dividing the data processing into successive, conceptual stages avoids any significant alteration of the programming code for different datasets or visualization goals, a hazardous process that possibly affects the application robustness and overall fault-tolerance. Generally, only minor changes within the structure layer are required, allowing the application designer to test different parameters and rules that influence the metaphor behavior instead of having to consider the inner logical workings of the program itself. In future versions, such adaptations might be accomplished through a user-friendly interface that facilitates real-time behavior rule alterations without having to alter the source code.

3.5.2. Update

The core infoticle metaphor concept consists of a direct mapping mechanism between a unique data object and an infoticle. For time-varying datasets, the data values of such a data object change over time. In practice, this means that a database table contains several entries of the same data identifier, at many different points in time.

A *data update* can be imagined as the sequential change of data values to a successive timeframe for a specific data object, which is uniquely identified by a data object *identifier*. A *timeframe* has a specific predefined duration and consists of a starting and end point along a timeline. Figure 3.26 shows the fundamental difference between the *database timeframe* and the *application* or *simulation timeframe*.

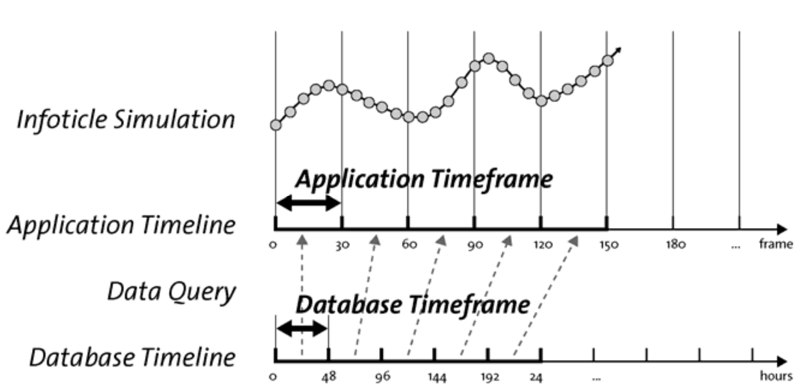


Figure 3.26. Application & database timeframe.

A *time update history* is the evolution of an accumulated sequence of data updates. The relative qualitative change of a single data update can be judged either in comparison with parallel updates of other data objects or with the time update history of the same data object.

– **Database Timeframe.** The database timeframe describes the duration in which the data that is streamed from the database to the visualization system needs to be positioned in time. It is typically measured in physical time units. Each data entry in a time-varying dataset contains a specific timestamp, which enables the system to order and visualize it in a simulated time. In practice, the visualization system queries the database for all data that is situated within the actual database timeframe. At every simulation timeframe step, the database timeframe is sequentially shifted to the next timeframe and new data is collected. Differences in the duration of the database timeframe results in the visual analysis of the time-varying data in subsequent time granularities, and thus in fundamentally different visual patterns. The database timeframe duration is also directly related to the quantity of the cached data, and therefore has a considerable effect on the application computing performance.

- **Application Timeframe.** The application timeframe denotes the rhythm at which the visualization simulation retrieves the next batch of data objects for the following database timeframe. In practice, it contains a specific number of frames after which the visualization system retrieves the new data objects of the next database timeframe. Each subsequent application timeframe corresponds to a unique, sequential database timeframe, and both are continuously updated in parallel. Differences in the duration of the simulation timeframe result in different infoticle movement speeds. As the infoticle trajectories need a certain amount of adaptation time, the resulting data patterns are not only faster, but also fundamentally different in nature.

The exact durations of both timeframes are typically defined at the visualization initialization, but can also be dynamically changed during the simulation itself. A detailed description of this time alteration process will be provided in the next section.

Determining the relationships between data object values and their evolution in time is not trivial. Mostly, it is unknown how, when and what quantities of the data are altered in time. In effect, the *frequency of reoccurring data objects* within a database timeframe determines two fundamentally different sorts of time-varying datasets, namely *singular* and *parallel sequential time-varying*. As will be shown, this dynamic characteristic directly affects how the infoticle visualization method processes updated data objects.

3.5.2.1. Singular Sequential

Some datasets consist of data objects that only change once within each database timeframe, resulting in a *one-to-one* relationship between data value changes and database timeframe. Whenever the database timeframe is sequentially updated, only one new set of values for a single data object exist. Datasets with such characteristics include *daily* closing stock market quotes, in which each company corresponds to a single closing price value for every day, as it is impossible for a company to receive none or multiple ‘daily’ closing prices. Consequently, the update mechanism for such datasets is relatively simple, as the *expected* data amount remains constant, and search algorithms can be optimized accordingly. Figure 3.27 shows how the infoticle metaphor updates the data objects for such datasets. When the simulation timeline has updated the actual application timeframe, the infoticle searches for a corresponding data object with new data values within the next database timeframe, and removes the old one.

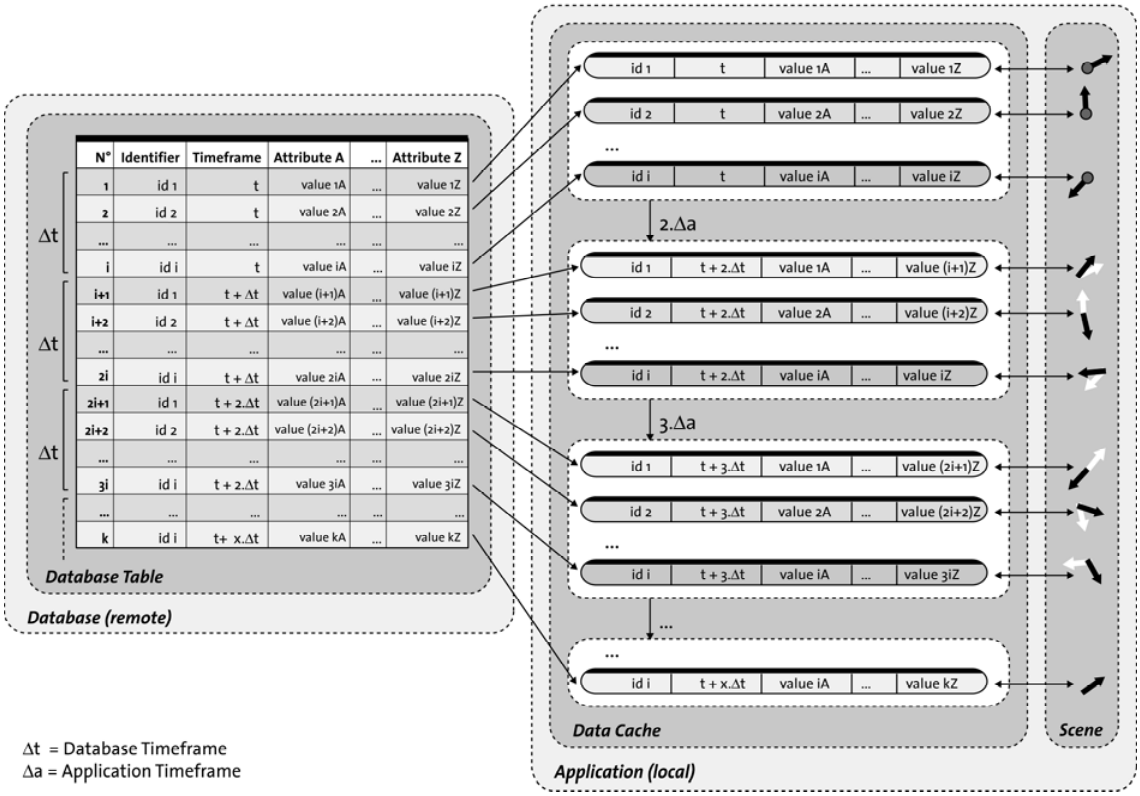


Figure 3.27. Singular data update.

3.5.2.2. Parallel Sequential

Most dynamic datasets contain a *one-to-many* or *one-to-null* relationship between a specific database timeframe and a unique data object. Examples of such *non-singular* or *parallel sequential time-varying datasets* include *live* stock market quotes that typically fluctuate more than once every second, and can contain companies having multiple price changes at unexpected points in time. Theoretically, such parallel sequential datasets can be converted into sequential datasets if the database timeframe is chosen small enough to exclude multiple data entries. However, reducing the database timeframe generally increases the amount of necessary successive database queries dramatically and produces very frequent streams containing only small amounts of data value changes per application timeframe. Often the database does not contain a data object update for one or more successive database timeframes. As a result, the data values of the corresponding infoticle stay constant during the respective application timeframes. Parallel sequential data complicates the data retrieval and update process, as the system is unaware of and is unable to predict the reoccurring frequency of each data object within a certain database timeframe. As will be explained in the next section, the optimal timeframe duration is directly dependent on the expected data quantities to be processed by the database communication and the allocated computer memory, and the desired level of detail of the time-dependent trends within the data.

In practice, this means that multiple data entries of a single data object cannot be avoided if the database timeframe duration is larger than the smallest *time granularity* of the database entries. Therefore, the infoticle metaphor needs to handle effectively all sorts of datasets. Three different methods have been developed that enable real-time parallel sequential data processing.

- **Sequential Simulation.** The system visualizes all available data objects sequentially within the active database timeframe. This means that, at each single frame, each infoticle checks whether there are still available data objects with its corresponding identifier stored inside the data cache. If so, the data values are updated, always resulting in maximally one data update, per data object, per frame.

  This approach might seem attractive at first for its apparent simplicity, yet shows some important disadvantages: the update needs to be fast enough to process all occurrences within each application timeframe, resulting in a rapid succession of almost non-perceivable patterns. Furthermore, the last available data entry of the data object within the actual timeframe has a clear advantage and will be visualized during a relatively longer time period. The defined application timeframe might be insufficient as one cannot predict how long the system needs to process all the parallel data objects. Dynamically adapting the application timeframe is not an effective solution, as extreme frequent data entries might monopolize long visualization simulation sequences. At the same time, this algorithm is highly calculation efficient as the system never needs to update more than one data object, per infoticle, per frame.

- **Average Calculation.** When visualizing quantitative data, the system calculates the average of all data objects that are *present in parallel* within the actual database timeframe, as demonstrated in Figure 3.28. For instance, this approach produces the daily average price of a live stock market quote simulation when the database timeframe equals one day, and the database includes hourly or minutely data entries. The average numerical data value $c_{average}$ of data attribute c with a frequency f_c within the database timeframe $\Delta t_{database}$ is calculated as follows.

$$c_{average} = \frac{\sum_{i}^{f_c} c_i}{f_c} \rightarrow \left(c_i \in \Delta t_{database}\right)$$

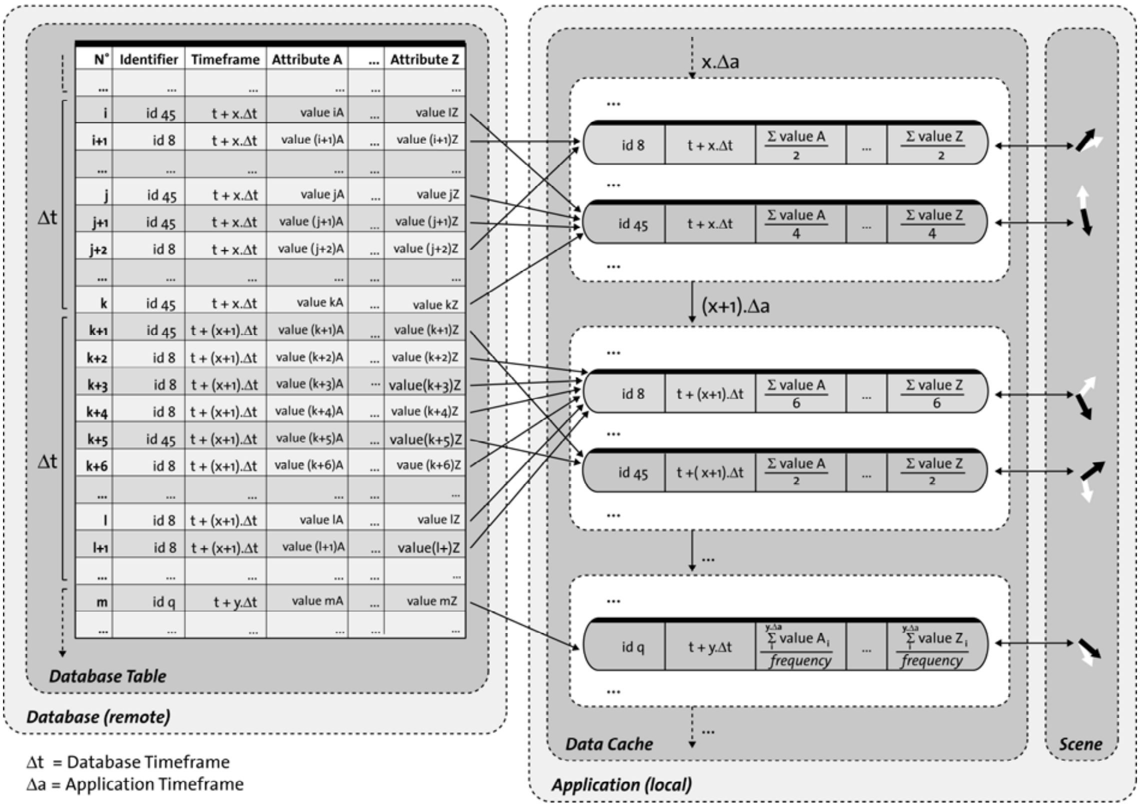


Figure 3.28. Parallel data update.

For unstructured (e.g. textual) data attributes, however, a mathematical average cannot practically be calculated. Yet, the data values reoccurrences within a database timeframe can effectively be visually expressed by the average force calculation, which takes into account the frequencies of unstructured data values, and therefore functions as a visual representation of the parallel sequential average. First, all frequencies of each retrieved data value c_i that correspond to a conditional force value c_x of a force F_x with position $\vec{F}_x$ within the timeframe $\Delta t_{database}$ need to be determined. An average force (not time-averaged) for an infoticle world with n forces is calculated as follows.

$$c_i = c_x \Rightarrow f_x = f_x + 1 \rightarrow \left. \begin{pmatrix} c_i \in \Delta t_{database} \\ x \in [0,1,..,n] \end{pmatrix} \right\} \Rightarrow \vec{F}_{average} = \frac{\sum_x^n f_x . \vec{F}_x}{\sum_x^n f_x}$$

Notably, the average calculation mechanism moves the processing strain towards the effort of looking up all corresponding data objects within a database timeframe during a single application timeframe. Checking every entry within a memory array for possible data similarities is a very calculation-intensive process that slows down the application considerably when the data quantities within the database timeframe are high, or the condition needs to check a high amount of non-numerical values. As will be demonstrated later, several performance optimizations are possible, for instance by spreading the search process over the whole application timeframe duration.

- **Frequency Summation.** The system counts the number of times that a certain data object occurs within the actual database timeframe and stores it in an extra data object attribute. This number is then used to steer the infoticle behavior by mathematically changing an infoticle attribute (e.g. speed, direction) accordingly, or is represented as an extra data dimension by one of the visualization tools (e.g. trace ribbon width). Alike the average calculation method, the system needs to search for and test all data object occurrences within the data cache for each infoticle in the scene.

  As will be demonstrated in Chapter 4, most infoticle systems typically use a combination of the average calculation and frequency summation procedures.

3.5.2.3. Data Cache

As the infoticle behavior is influenced by the change of its corresponding data object, it should be able to access the data object values efficiently at all times. A live retrieval of the individual data objects from the database for each infoticle within the scene is too time-expensive, as this approach evidently requires considerable computing resources for both network communication and database querying. Instead, a local solution is preferred, which stores the infoticle objects together with their corresponding data objects within the shared computer memory range. To optimize this process, the data objects that comply with the *next* simulated timeframe are already fetched beforehand to the local machine by a *parallel* data querying and storage process. These data objects are accumulated within a specific shared memory *cache* that is accessible by both the collection layer and the structure layer. The exact mechanisms of this concept are explained in Section 3.7.3. Infoticle – Implementation – Data Processing.

3.5.3. Time

The infoticle metaphor is fundamentally dependent on the conceptual relationship between time and space, and its simulation within a virtual world. Research in cognitive science has shown that the effectiveness of animation is enhanced when users are able to directly manipulate the motion attributes. This requirement includes the ability to freeze, rewind and forward the simulation at different speeds.

3.5.3.1. Analysis States

The core of the infoticle metaphor is dynamic in nature, and relies on the perception of both static and dynamic visual patterns. Both approaches offer users the ability to perceive and interpret the visual representation in fundamentally different contexts.

- **Dynamic State.** The characteristics of the continuously and dynamically altering representation enable a visual analysis of patterns that emerge during the successive update of data values. Users are able to observe time-varying differences in *densities*, *directions* and *odd-performing* infoticles that behave different from the rest. They can detect dynamic visual patterns that resemble those in scientific visualization applications. For instance, one can categorize the dynamic behavior of infoticle clusters as *chaotic* or *laminar*. However, the constant visualization animation implies that users are attentive at all times, so that sudden events are always spotted and time-limited patterns are recognized before they disappear. For that reason, the infoticle application lets users interact intuitively with the time simulation, enabling them to manipulate both the time direction and speed. For instance, users are able to freeze the representation, rerun certain historical events in slow-motion or skip unimportant time sequences at higher speeds.

– **Static State.** An infoticle visualization can also be analyzed in a motionless state. When the visualization is frozen, clusters of adjacent infoticles can be easily detected and compared, and the formal language of the static traces and shapes can be interpreted. This state facilitates a more detailed data exploration, as infoticles are more easily selectable and the text labels more effectively readable. A static infoticle representation does not fundamentally differ from normal force-directed visualizations as the different infoticle distances represent degrees in informational similarity. In addition, the proximity between individual infoticles and the forces can be compared, whereas the infoticle length and direction denote the current force dependency. Such a static representation also visually resembles that of a large, three-dimensional scatterplot or Starfield visualization (Shneiderman, 1994), although no meaningful relationship exists between the spatial Cartesian coordinate positions and the represented data values.

3.5.3.2. Simulation States

Time simulation interaction is a fundamental infoticle metaphor feature. Because the infoticle application simulation needs to be fully interactive, the infoticle trajectories cannot be pre-computed. This restriction results in the live caching of the queried data and some real-time data attribute alteration algorithms. As shown in Figure 3.29, the application time direction can have four conceptually different states.

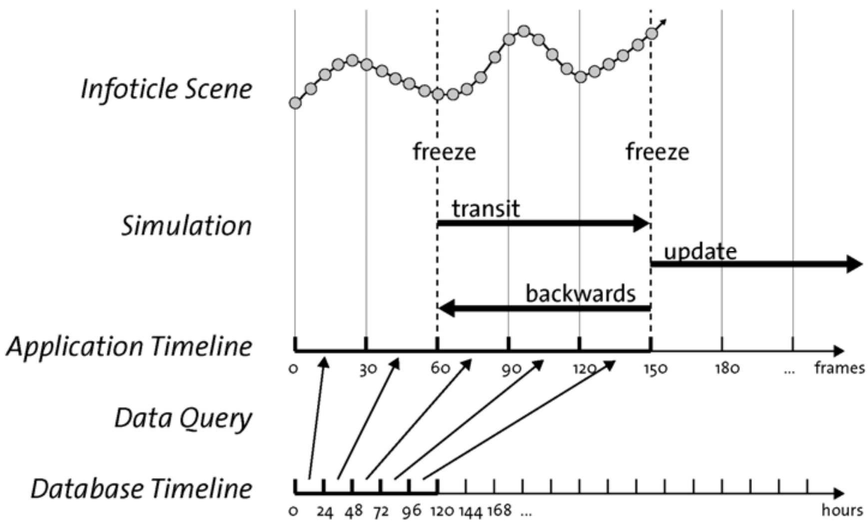


Figure 3.29. Time simulation states.

– **Update.** During the update state, the data is queried and cached at predefined application timeframe intervals. Consequently, the infoticle data object is updated and the corresponding infoticle attributes are altered, resulting in a specific spatial behavior. Normally, this data caching and updating process is performed at the maximum calculation speed possible, as the live simulation of the infoticle trajectories is very computing-intensive and difficult to optimize.

- **Backwards.** During the backwards animation state, the system fetches and linearly interpolates the coordinates stored in the individual infoticle history list. As a result, each single infoticle moves approximately along the path it followed before, as shown in Figure 3.30. The data caching has been stopped, and the system calculation load is reduced to a relatively simple interpolated particle animation along fixed control points. This is a calculation-cheap process, so that the simulation speed can be altered relatively easily by changing the *interpolation granularity*, which in turn corresponds mathematically to the application timeframe duration.

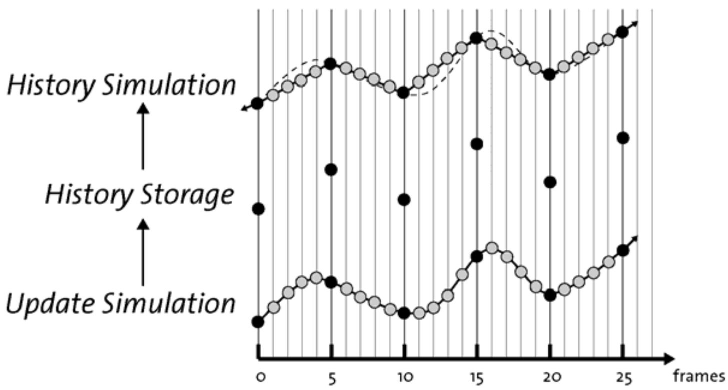


Figure 3.30. Trajectory interpolation.

- **Transit.** When the current system state is animated before the last update time because of a previous backwards movement, the system first needs to be brought back into the latest reached update state again. Consequently, the transit state, like the backwards state, is a result of interpolating coordinates retrieved from the history lists. The transit simulation speed can be easily altered, similarly to the backwards state. At the end of the transit state, the system switches automatically to the update state. Figure 3.30 shows how the system linearly interpolates the spatial infoticle coordinates between the history positions originally stored during the backwards and transit time direction. As a result, the spatial infoticle trajectory is closely approximated in time.
- **Freeze.** During the freeze state, all animations are stopped, and only the interface and interaction processes are executed. The processing load is thus reduced to a minimum, enabling rapid and direct interaction with the user, who is then able to focus on static text information and the analysis of static, simulation-independent spatial patterns.

One should note that, in contrast to the application timeline, the updated database time direction is always forward: whenever the application time is not updating, the database querying and data caching is halted until the visualization simulation has reached back the latest update state. Only then will the database time and the application time be exactly synchronized again. The backwards and transit simulation thus does not rely on live or cached data, but instead uses the data stored within the history lists to accurately visualize the application time direction.

3.5.3.3. Timeframe Size

The determination of the exact database and simulation timeframe durations is an important application design decision that has direct implications on the resulting visualization patterns and the required computing process performance.

– **Database Timeframe Alteration.** Each data entry in the database contains a certain timestamp, which enables the system to order and visualize the whole dataset in time. In practice, the visualization system queries the database for all data that is situated within a certain database timeframe. At every update step, this timeframe is sequentially shifted to the next timeframe and new data is collected.

The database timeframe is typically measured in *physical time units*, such as seconds, minutes, weeks, years, etc. The exact timeframe size is dependent on the time-varying dataset characteristics and the desired data visualization granularity. Ultimately, its duration is specified by the application designer who determines an ideal equilibrium between the desired detail of time-varying analysis, computing performance and estimated application execution time. Choosing a long database timeframe often results in huge database subset streams that are time-intensive to communicate through the network and require large computing resources on the local machine to process. However, specifying too small timeframes might cause a large amount of database queries, returning empty or sparse database subsets. The resulting widely distributed data changes are difficult to spot, producing a long and visually boring time simulation. Resizing the database timeframe enables the analysis of data patterns at different time scales, but also influences the total time needed to represent the complete dataset.

Normally, the exact database timeframe duration is fixed at the application initialization, as dynamic changes during the visualization simulation result in fundamentally different patterns that are difficult to comprehend and are hardly comparable. Figure 3.31 shows how database timeframe alterations cause non-predictable spatial trajectory adaptations, mainly because the infoticles receive different durations to adapt to the continuous influences of the various tools within the scene.

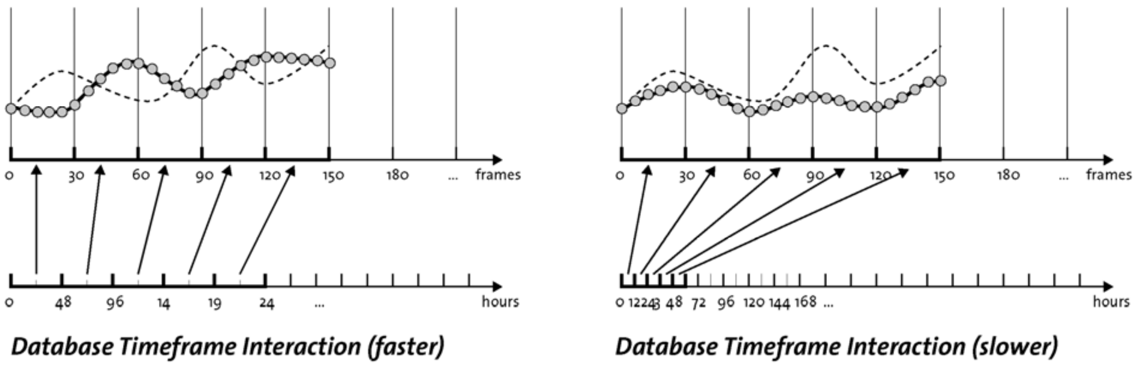


Figure 3.31. Database Timeframe Interaction.

– **Application Timeframe Alteration.** The application or simulation timeframe denotes the exact number of frames at which the visualization system retrieves the new data objects of the next database timeframe. Each subsequent application timeframe corresponds to a unique, sequential database timeframe, and both are updated in parallel.

This timeframe duration is determined at the visualization initialization, and normally stays unchanged during the update state. This rule ensures that all visual patterns result from the same update speed and thus can be easily compared in both space and time. Although the application timeframe duration is always fixed during the update state, it can be dynamically changed by users during both the backwards and transit state, as these only require data from the fixed history lists and thus do not alter the data representations themselves. Figure 3.32 demonstrates how application timeframe alterations produce faster or slower movements along equal spatial paths in the backwards and transit state. In practice, dynamic changes of application timeframe durations thus enable users to skip certain time sequences rapidly, or to analyze potentially interesting events in slow motion.

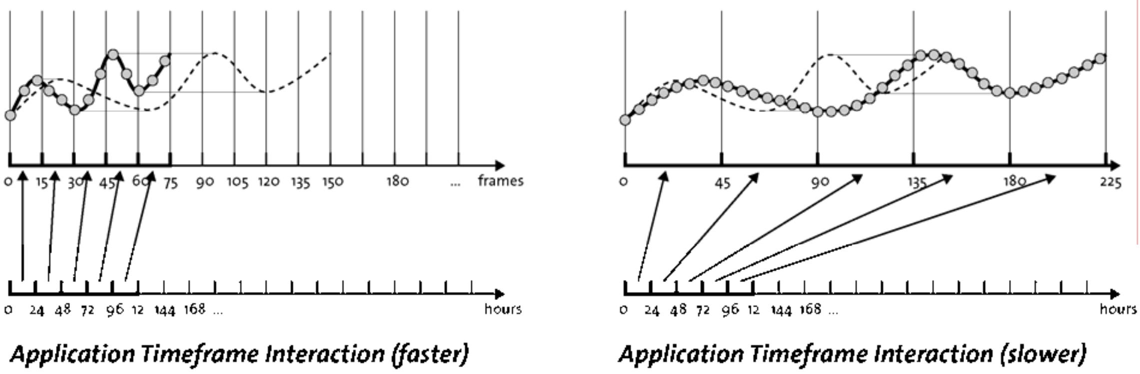


Figure 3.32. Application timeframe interaction.

3.5.3.4. Future

One might wonder what would happen if the infoticle simulation ran beyond the last available database timeframe. In fact, this would mean that the infoticles are visualizing the future. Although practically impossible at first sight, this theory would be achievable when certain learning algorithms altered the data values of the individual infoticles based upon the data stored within the history lists. In addition, it might be theoretically possible to alter the force strengths, instead of the infoticles. This scenario would be especially suited to test certain possible future data trends. In practice, the system would first need to track both the quantities and attraction durations of all infoticles it influences, and mathematically project this experience onto its own force strength. In effect, adapting force attraction strengths with dumb infoticle behavior would result into a reasonably correct representation of future global trends. However, these visual patterns are only valid on a macro-scale, and are not directly related to individual data values.

3.5.4. Behavior

Although both particle systems and behavioral animation were originally designed for modeling and animation applications, the combination of these two concepts provides a synergy with highly promising contributions in the context of information visualization. It is known that, given correct initial conditions, and combined with internal relationships, particle systems can be simulated over time, and can convey complex behavior (Tonnesen, 2001). Accordingly, Table 3.1 shows the different traditional methods that generate particle-like animations through techniques that are not based upon predefined outcomes, and thus are able to determine so-called *emergent behavior* (Parent, 1998). The following sections will explain how the data-driven particle method enhances the infoticle animations with flocking behavior and local rules that drive the real-time interpretation of data updates.

Animation	Participants	Intelligence	Physics
particle	*many*	*none*	*yes*
flocking	*some*	*some*	*some*
behavioral	*few*	*high*	*no*

Table 3.1. Particle animation characteristics.

– **External.** The previous sections have described that each infoticle represents a unique data object. Data attributes are updated with new time-ordered values retrieved from the database at a rhythm that is synchronized with the application timeframe. Simultaneously, various tools within the scene, such as filters and forces, alter the attributes of those infoticles that represent matching data values. Consequently, data values changes will result in the dynamic *switching* of individual tools that influence the infoticle, so that its trajectory represents the history of spatial force attractions.

– **Internal.** Next to the manipulation by these external tools, specific *internal* rules generate more subtle spatial behaviors. The data value updates accumulated from the collection layer generate specific infoticle behaviors within the structure layer. As a result, the updated data values are mapped onto various particle motion features, hereby directly translating the dynamic nature of time-varying information. This section will describe in detail the behavior rules that facilitate the translation of data changes into different alterations of infoticle attributes. Accordingly, Figure 3.33 demonstrates how an infoticle trajectory is influenced by a possible behavior change triggered by a data object update.

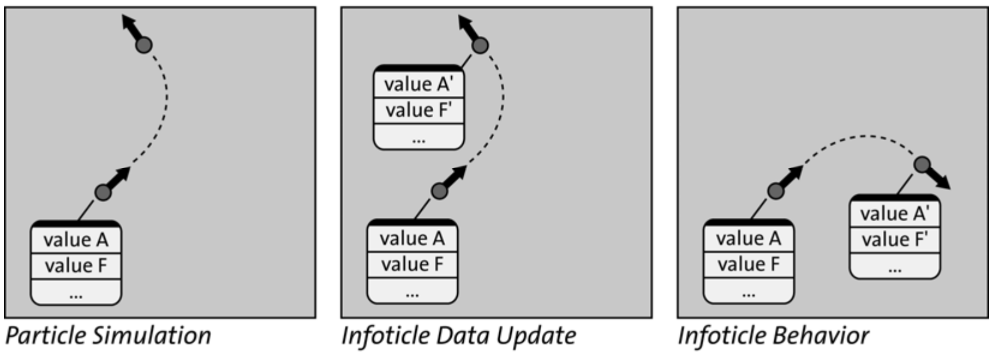


Figure 3.33. Behavior influence.

The specification of this behavior system is a result of many *trial-and-error experiments*. Through a process of empirical selection, specific dynamic behaviors have been selected that are easily and intuitively interpretable. For instance, a behavior that is mainly based upon speed or mass alterations initially and seemingly results in understandable circular and elliptical trajectories, similar to those used in *mass spectrometry*. However, it was quickly discovered that the speed adaptations themselves could not be effectively recognized as they produce chaotic trajectory alterations. This randomness is caused by the large dependency on the relative position within the elliptical trajectory because of the natural change of speed, better known as *Kepler's 2nd law*. Another observation showed that infoticles at great distance from forces or with relatively large speeds often are only slightly affected by force attractions, so that these infoticles first need to be artificially redirected towards the force center in order to guarantee an effective attraction.

In general, behavior rules need to generate visual patterns that are interpretable by users and can be intuitively linked to time-varying data update characteristics. Similar considerations led to the Space Galaxy and Boid behavior specifications.

3.5.4.1. Galaxy

The following behavior rules are valid for each infoticle and are executed at each single frame. They result in a global visual behavior that resembles a *space galaxy*, comprising different types of motions and trajectories that visually relate to various astrophysical phenomena.

- **Tool Influence.** Adapt the spatial infoticle attributes to the tools present in the scene. In practice, this means that an infoticle needs to check at each frame all filters and forces for possible data value similarities. Furthermore, each infoticle needs to calculate a new time-averaged force. Subsequently, its current speed and direction vector is adapted to the new average force attraction vector.
- **Drag.** Gradually reduce speed and color brightness. This rule enables the spatial visualization of data values having different update frequencies. For instance, when a data object is not updated for a longer period of time, the infoticle will slow down or almost come to a standstill, and visually be harder to detect because of its minimal brightness. In the context of vector mathematics, an almost immobile infoticle is much more sensible to spatial attractions as its speed vector is relatively small compared to possible external influences. The drag rule is limited to the reduction of the magnitude of speed s with a weight factor w, and does not influence the direction of the *normalized vector* $\vec{v}$. However, this speed alteration will possibly have a direct repercussion on the future trajectory of the infoticle, as external force attraction has a larger impact on slow infoticle velocities.

$$\left.\begin{aligned} \hat{v} &= \frac{\vec{v}}{|\vec{v}|} \\ s_{old} &= \|\vec{v}\| \end{aligned}\right\} \rightarrow \vec{v}_{new} = \hat{v}.(s_{old}.w_{drag}) \Leftarrow \begin{bmatrix} 0 < w_{drag} < 1 \\ \vec{v}_{new} > \vec{v}_{\min} \end{bmatrix}$$

- **Orbit.** If the distance to the average force becomes too small, start orbiting around it. This rule avoids that the infoticle comes too close to the force center, which would result in dramatic speed increases directing the infoticle far outside the visualization.

The infoticle orbit implementation is based upon the well-known *centripetal force* definition. A centripetal force keeps any object in circular motion and points in the same direction as the acceleration. The centripetal force F_c is equal to the mass m times the velocity v squared, divided by the radius r of the circle. Consequently, if the attraction strength of a force is known, the required velocity to bring an object in a permanent circular trajectory is calculated as follows.

$$F_c = \frac{m.v^2}{r} \Rightarrow v = \sqrt{\frac{F_c.r}{m}}$$

An infoticle is influenced when the distance d between the infoticle and the force is smaller than the predefined *force catch range* D_F. In this case, an influence direction c is calculated orthogonal to both the vector connecting the force center and the infoticle, and the current infoticle direction. Next, the infoticle direction is altered to a collision avoidance course. Consequently, its speed must be exactly adapted so that the infoticle will move in a uniform circular trajectory around the force. Therefore, the normalized influence vector $\hat{c}$ is multiplied with the necessary centripetal speed magnitude, which is dependent on the force attraction strength $F_{average}$ and the infoticle-force distance d.

$$d = \left\| \vec{F}_{average} - \vec{p}_{old} \right\| < D_F$$

$$\vec{c} = \left| \vec{F}_{average} - \vec{p}_{old} \right| \times \vec{v}_{old}$$

$$\vec{v}_{new} = \hat{c}.\sqrt{\frac{F_{average}.d}{m}}$$

– **Update Data.** Check whether the data values have changed. If so, perform the following actions.
  – **Accelerate.** Increase speed.
  – **Brighten Up.** Increase brightness.
  – **Revaluate.** Calculate new average force.
  – **Redirect.** Direct towards the new average force.

The sudden visual change caused by the brightness enhancement, the spatial direction variation and the speed increase helps users easily *spot* the actual data changes within the data visualization simulation. In addition, the speed increase compensates for the second behavior rule.

Next to these rules that are applied to each single infoticle, the visualization system keeps track of the global logical structure of the dynamic simulation and synchronizes the timelines of all infoticle emitters that are present within the scene. In a pace synchronized with the application timeframe rhythm, it determines the next database timeframe and, for each infoticle system, queries the database for all the data objects available within it. While the visualization continues to smoothly simulate the different motion behaviors, the data acquisition process awaits the resulting subset of data and stores it within the shared data cache for later use. At the tempo of the application timeframe, the system performs the following tasks (Figure 3.34).

– **Query.** Query, wait, retrieve, order and cache the data objects of the next database timeframe within the local data cache. Share this memory range with the infoticles.

– **Update.** Save and update the historical data of each infoticle in its respective history list. Order this list in time by putting new data values at the top, and remove the oldest, and thus last, data entry when the list becomes too long.

– **Switch.** Check for each infoticle whether one or more available data objects exist in the cache for the active database timeframe. If so, use one of the parallel sequential dataset solutions, switch to the new (or derived) data object, store the values of the old data object in the history list and remove the old data object from the shared memory.

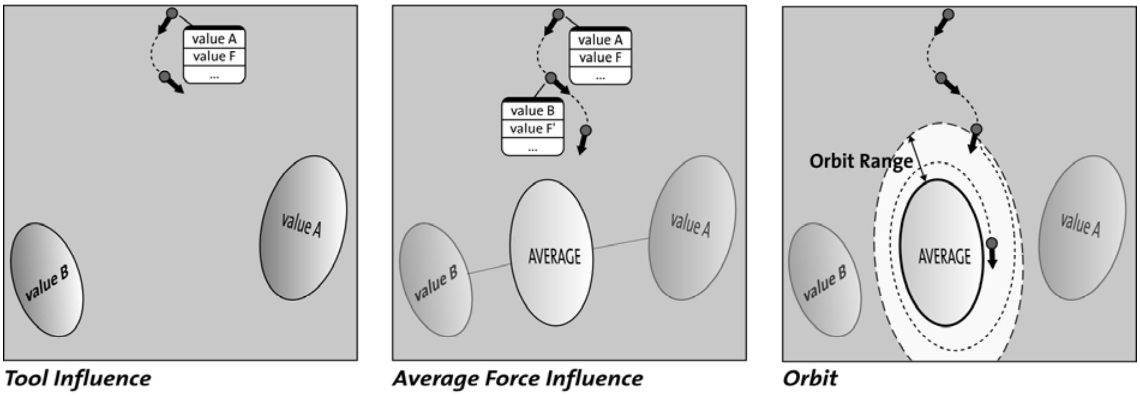


Figure 3.34. Space galaxy behavior.

3.5.4.2. Boids

More sophisticated behavior rules systems depend on the state of the surrounding environment and are able to generate more complex patterns. One popular approach to produce animation is to let objects obey certain rules that govern their actions, resulting in so-called *emergent behavior*. A behavior is called *emergent* when it arises out of the *interaction* between subunits in the system that do not show this property individually, so that the resulting behavior of a whole group transcends what the members have been explicitly programmed to do. In the fields of artificial life and biomimetics, emergent behavior is generally generated through so-called *local rules* that are valid in parallel for a large number of identical entities. These rules are *local* in two senses: each individual member contains its own set of rules and the future state of a member only depends on its neighbors. Consequently, the local rule approach differs from the simple Space Galaxy behavior rules listed above in its fundamental dependence on the surrounding infoticles. By updating the states of all members in parallel, a complex and unpredictable system can emerge. Examples of such systems include the Game of Life and the artificial simulation of collective intelligent behavior of ants and bees. Heuristic-based particle behaviors include follow, flocking, hunting, eating, life, death and spawning. In contrast, physically-based particle behaviors include the forces of gravity, wind, viscosity, attraction, repulsion and noise.

The oldest example of complex emergent behavior conveyed by particles was introduced by Reynolds, who successfully modeled the behavior of blackbirds moving in a flock (Reynolds, 1987). The movement of these particles, called *boids* (short for bird-objects), are determined by both internal as well as external factors. Reynolds' hypothesis was that boids should act as *agents*: they are situated, viewing the world from their own perspective rather than from a global one. He used the observation of real flocks and flock behavior research to come up with the three primary needs of a boid. These rules need to be executed at each point in time, for each individual boid.

– **Collision Avoidance.** Avoid collision with other boids nearby.

– **Velocity Matching.** Match the speed and the direction of boids nearby.

– **Flock Centering.** Attempt to stay close to boids nearby.

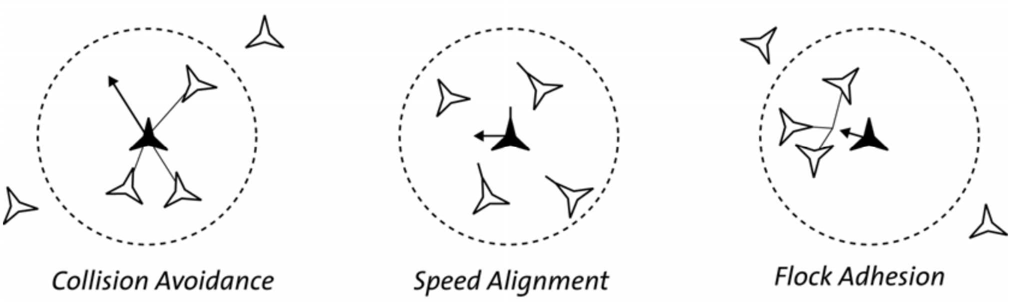


Figure 3.35. Boid Behavior Rules.

The first rule defines the tendency to steer away from an imminent impact and avoids the boids to cluster too closely. The concept of velocity matching pushes boids to move in the same direction so that, as a group, they seem to have a common goal. The third rule steers a boid near the center of the flock, which is the perceived center of gravity of the other boids nearby. Consequently, boids in the center will feel no pull outwards, whereas boids in the periphery will be deflected. According to Reynolds, instead of simply averaging these influences, a *prioritized acceleration allocation* strategy should be used. This approach allocates priority according to the order of controllers and normalizes values when controllers receive less than what was requested. In the end, a weighted average is used to compute the final velocity vector.

As one of the emergent effects, boids are capable to avoid obstacles, to split in subgroups and to regroup afterwards. Furthermore, boids show the ability to adapt smoothly to unexpected external situations, a phenomenon that surely has great potential for time-varying information visualization purposes. Accordingly, the infoticle metaphor has extended the traditional boid metaphor with the dynamic information visualization approach of data-driven particles. Subsequently, the traditional boid behavior rules received one extra internal relationship that relates to the data objects represented by infoticles.

– **Data Similarity.** Attempt to stay close to those boids with similar data values.

– **Data Dissimilarity.** Attempt to avoid stay away boids with dissimilar data values.

In practice, these extra rules indicate that infoticle boids are attracted to each other when their respective data values show some sort of *data similarity*. The *data dissimilarity* repulsion reinforces this data clustering phenomenon. Following this approach, the directional speed vector of an infoticle boid A is calculated as follows. Each new directional vector of boid A is an accumulation of the attractive or repulsive forces between boid A and all its neighbors, whenever specific distance clauses are fulfilled. For instance, the collision avoidance rule is only invoked when two boids are relatively close to each other, whereas the velocity matching rule generally has a wide range of influence.

$$\|\vec{p}_i - \vec{p}_A\| \le d_{CA} \Rightarrow \vec{v}_{CA} = \sum_i \frac{|\vec{p}_i - \vec{p}_A|}{\|\vec{p}_i - \vec{p}_A\|}.$$

$$\left.\begin{array}{l}\|\vec{p}_i - \vec{p}_A\| \le d_{VM} \\ \|\vec{p}_i - \vec{p}_A\| > d_{CA}\end{array}\right\} \Rightarrow \vec{v}_{VM} = \sum_i \vec{v}_i$$

$$\left.\begin{array}{l}\|\vec{p}_i - \vec{p}_A\| \le d_{FC} \\ \|\vec{p}_i - \vec{p}_A\| > d_{CA}\end{array}\right\} \Rightarrow \vec{v}_{FC} = \sum_i |\vec{p}_i - \vec{p}_A|$$

$$\left.\begin{array}{l}\|\vec{p}_i - \vec{p}_A\| < d_{DS} \\ similar\ data\end{array}\right\} \Rightarrow \vec{v}_{DS} = \sum_i |\vec{p}_i - \vec{p}_A|$$

$$\left.\begin{array}{l}\|\vec{p}_i - \vec{p}_A\| < d_{DD} \\ dissimilar\ data\end{array}\right\} \Rightarrow \vec{v}_{DD} = \sum_i |\vec{p}_i - \vec{p}_A|$$

$\vec{p}_i, \vec{p}_A$ are the position vectors of boid *i* and neighboring boid *A* respectively. d_{CA}, d_{VM}, d_{FC} and d_{DS} are the different influencing distances or catch ranges. Whenever the length of an accumulation vector is too large, it becomes normalized.

$$\|\vec{v}_x\| > 1 \Rightarrow normalize(\vec{v}_x) \rightarrow x = \{CA, VM, FC, DS, DD\}$$

Separate weights reflect the importance of the different behaviors. Typically, the collision avoidance weight is very high, whereas flock centering is minimized in favor of the data similarity influence. The exact specification of these distance and weight variables is determined after a process of empirical experimentation. Most behavior influences exist in internal, attractive forces, only collision detection and data dissimilarity denote a repulsive influence.

$$\vec{v}_A = -w_{CA}.\vec{v}_{CA} + w_{VM}.\vec{v}_{VM} + w_{FC}.\vec{v}_{FC} + w_{DS}.\vec{v}_{DS} - w_{DD}\vec{v}_{DD}$$

$0 < w_{CA}, w_{VM}, w_{FC}, w_{DS}, w_{DD} > 1$ are weights applied to Collision Avoidance, Velocity Matching, Flock Centering, Data Similarity and Data Dissimilarity behaviors respectively. $\vec{v}_A, \vec{v}_{CA}, \vec{v}_{VM}, \vec{v}_{FC}, \vec{v}_{DS}, \vec{v}_{DD}$ denote the directional speed vectors.

Interestingly, this infoticle boid concept was originally invented without the knowledge of a quite similar approach introduced by the British Telecom Artificial Life Group in 1998, which uses the metaphor of swarming fishes to represent static similarities in interest of groups (Proctor and Winter, 1998). Primary differences exist in the following issues.

– **Time-Varying Dataset.** Although time-varying datasets are mentioned very shortly once, the previous approach seems to focus solely on static datasets. In fact, one might note some discrepancy between the nature of boids that are continuously moving (and thus constantly reforming spatial clusters), and the representation of static data. In contrast, infoticles are able to adapt to rapid and even chaotic data update sequences.

– **Real-Time Evaluation.** The infoticle method does not need a pre-calculated similarity matrix denoting the relationships between all individual data entries, but determines the data similarity during execution time. Consequently, real-time datasets and different dataset typologies can be represented.

– **Algorithm.** In contrast to the other approach, the infoticle boid approach has not removed the general Flock Centering rule because it enables a more effective control over the global and time-varying coherence of the infoticle collection. This algorithmic difference between the two methods is most probably caused by the continuous data update process, which evidently requires more intrinsic, dynamic stability and thus more organizing power over the whole behavior generation.

– **Dissimilarity Repulsion.** Two, instead of one, extra factors extend the original boid algorithm by considering data dissimilarity as a complete, individual influence. Alternatively, dissimilarity could not be evaluated at all, or be merged into a single data influence vector. However, adding this influence separately enforces the clustering and especially the declustering tendency of boids.

3.5.4.3. Patterns

These two different behavior generation procedures lead to the emergence of different motion typologies that are easily recognizable. The resulting dynamic cues and their conceptual relationship to the nature of time-varying datasets will be explained in detail for each individual application prototype in Chapter 4.

3.6. Interface

An *interface* typically denotes all information channels that allow user and computer to communicate. *Interface design* focuses on the aspects of interaction in a user-centered and task-oriented way, and hereby attempts to maintain a specific common ground between application and user.

The following paragraphs describe the design philosophy behind the infoticle interface paradigms. In fact, designing effective interfaces for immersive, stereoscopic environments is still poorly researched. Many of the specific results and guidelines that are offered by *Human Computer Interaction* (HCI) practitioners (Hix and Hartson, 1993) do not apply directly to immersive virtual reality applications. Mostly, traditional interface research focuses on issues that are placed within a fundamentally different technical context, and do not have to cope with difficulties of interaction within three-dimensional space, the fundamental dissimilarities in both input and output devices, the slow system response times, and so on. A few researchers have attempted to standardize three-dimensional interfaces by evaluating interaction and navigation metaphors that allow for effective usability in a wide range of virtual reality environments (Hinckley, et al., 1994, Bowman, 1999, Bowman, et al., 2001). However, experience has proven that standardization often is unnecessary and interaction features should instead be optimized for particular tasks in specific domains.

3.6.1. Design

Developing interfaces for three-dimensional virtual worlds fundamentally differs from traditional screen interaction design, so that the potential contribution of two-dimensional graphic and interaction design research is limited. In fact, an immersive visualization space represents an overall dimensionality that can be compared to that of everyday physical reality. Consequently, the visualization metaphor has to generate an effective visualization that is adapted to the *human scale*, rendering simple measurement adaptations of currently existing desktop visualization applications impossible. Virtual reality application designers need to be aware of the necessary dimensionality of the virtual representation space to create data patterns that are easily noticeable. Furthermore, they have to take care of perception details such as text label readability and navigation simulation velocity. The design of a three-dimensional interface for immersive virtual reality has to take into account the following aspects.

– **Resolution.** Although presented on a physically huge scale, the typical virtual reality display resolution equals that of a normal computer desktop screen. In effect, 1024 by 768 pixels are blown up to a display space of about 3 by 2.3 meters, resulting in a surface area of ± 9 mm^2 per pixel. Consequently, the smallest easily readable text labels need to be presented within a 10-pixel high screen space, a dimensional solution space that is surprisingly similar to older generations of mobile phone displays.

  The infoticle metaphor is a good demonstration of the limited capabilities of contemporary virtual reality technology to effectively display fine-detailed elements such as text. Some infoticle prototypes attempt to overcome this technical problem with a technique that can be compared to so-called *pilot heads-up* or *cockpit displays*, which render all text in equal size, irrespective of the distance the corresponding element is positioned in the three-dimensional representation space (Figure 3.36).

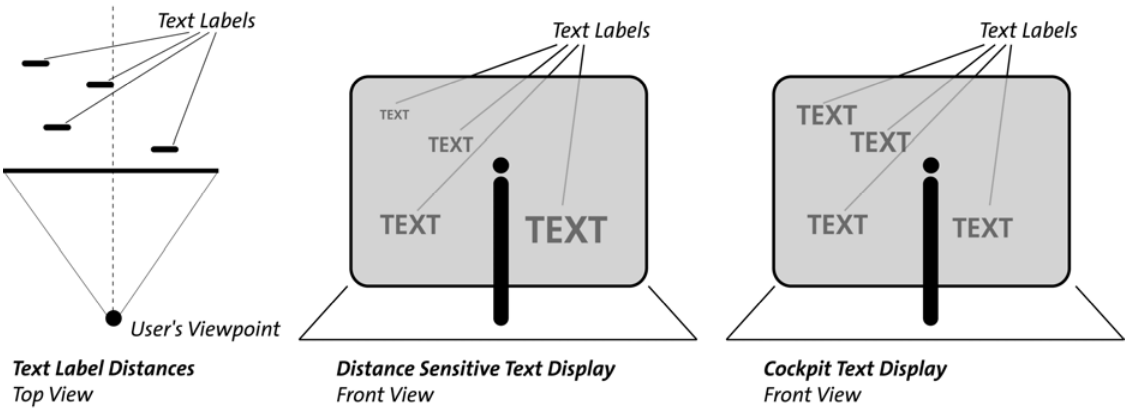


Figure 3.36. Infoticle text scaling.

  Other resolution-related artifacts can be detected when representing graphical elements with such physically large pixels. For instance, the display of fine lines or typefaces typically results in visually disturbing non-aliased jagging pixel rows. Consequently, virtual reality applications in general should not be based upon displaying high information densities per display surface area, as these cannot be effectively presented.

– **Text Directionality.** The implementation of three-dimensional interface design also differs significantly depending on the presentation medium. For instance, text labels shown in three-dimensional worlds displayed on flat displays, ranging from simple desktop screens to monolithic stereoscopic projection walls, are typically directed towards a fixed axis in physical space at all times. In contrast, Figure 3.37 shows how textual elements in immersive environments always should be pointing exactly towards the user's viewpoint to be readable on all surrounding screens. This phenomenon produces a dramatic increase in computing effort, as the three-dimensional directional vectors of each single text label in view have to be calculated at each navigational change, and this at each single frame.

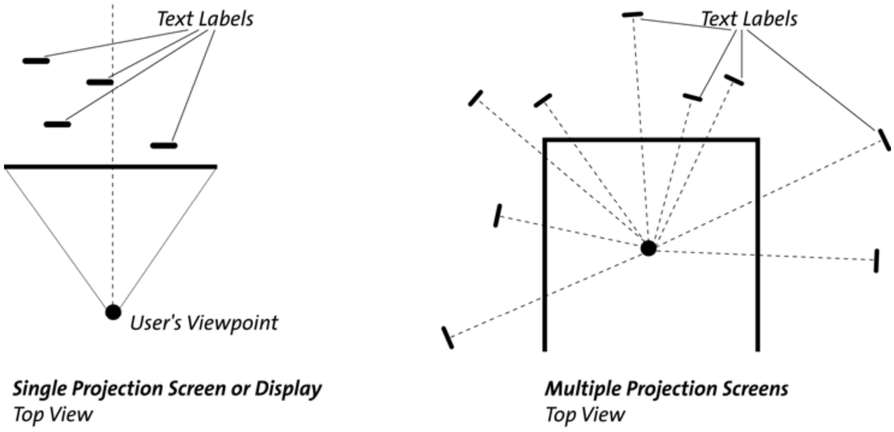


Figure 3.37. Immersive text presentation.

- **Interaction.** Virtual environments lack a unifying framework for interaction, such as the desktop metaphor for conventional operating systems. Using three-dimensional metaphors for interaction often results in real-world metaphors and many user experience issues, as the complete lack of haptics blocks the intuitive knowledge retrieval on how to manipulate objects from the world itself. In addition, placing menus or widgets within virtual environments breaks the illusion of being present inside a truly three-dimensional world.

  Instead, the infoticle method positions all interface elements within the scene itself. This approach reinforces the three-dimensional spatial awareness of users and increases the feeling of mental and physical immersion, generating an enhanced user engagement. Simultaneously, the infoticle application attempts to decrease the complexity of interface interaction by enabling direct manipulation of all elements within the scene. Instead of explaining users the distinction between static objects and interface elements, users are able to select, drag and rotate all visible objects. For instance, forces can be selected and dragged together, so that new infoticle constellations start to emerge. Infoticles can be pushed or pulled to another region for better examination or to test whether certain data objects belong together.

- **Cursor.** Traditional virtual reality cursors that facilitate object selection within the scene typically consist of a virtual human hand, a three-dimensional magic stick or a small collision-detecting sphere. They often are the source of many user frustrations as users have difficulty directing them precisely and efficiently within three-dimensional space.

  The infoticle system cursor comprises two blended circular icons that mimic the normal desktop interaction metaphor (Figure 3.38). These two circles have different distances in the depth direction and are placed on an imaginative straight line that starts at the user's hand and follows the point direction of the input device. As a result, elements within the scene can be selected by clicking a button when they are overlapped by the smaller, front cursor icon. The back icon represents the depth effect of the directional cursor. In effect, users are able to employ this interaction technique quite similarly to the desktop metaphor. Experience has shown the simple cursor metaphor to be very effective and to be appreciated by a large majority of users.

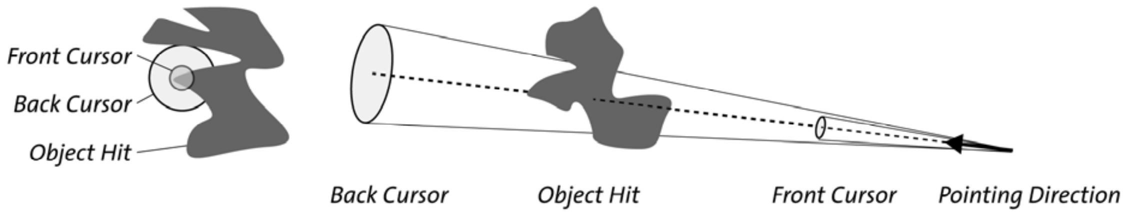


Figure 3.38. Infoticle system cursor.

– **Background Color.** Although it is probably the most typical and unappreciated commonality of most virtual reality applications, the solid black background color was primarily chosen for purely technical reasons. Experience has proven that bright background colors significantly amplify the mirroring reflections in the glass projection walls of the blue-c environment and increase the so-called circular hot-spot blending disturbance of the bright projector lamp through the projection screen.

– **Colors.** The choice of colors employed in the infoticle system is directly influenced by the insights from traditional information visualization guidelines in addition to global information-aesthetic motivations.

– **Icons.** As mentioned before, all interface tools are positioned within the scene. The graphical ‘icon’ design of the filter and force elements have been kept to a minimum, and do not convey an abstracted representation of their actions like traditional icons. Their front view is always directed towards the actual user’s viewpoint, similar to the billboard effect, so that their graphical shape can be effectively recognized at all times. In addition, they are the only static elements within the virtual world, and thus function as the main fix visual orientation cues when navigating through the data visualization. In addition, all icons are surrounded by large circular boundaries so that users are able to select them more easily with the virtual cursor, and require little precision effort to point and select smaller elements on greater distances.

– **Navigation Speed.** To preserve a believable feeling of presence inside immersive applications, the navigation speed must not be too fast to confuse ordinary users or too slow to make effective work impossible. In fact, the infoticle method uses the joystick on the input device to determine the navigation speed between certain predefined default thresholds.

3.6.2. Dimensionality

Representing interface elements by three-dimensional objects within the virtual scene is still a relatively unexplored approach. The infoticle user interface neither breaks the three-dimensional illusion nor occludes the virtual world with text- or icon-based menus, sliders or other widget elements. The chosen design principles also avoid the use of small, and thus minimally occluding, icons that are difficult to visually recognize or select (or *hit*) in three-dimensional space. Compiling interfaces out of three-dimensional elements that appeal to everyday familiarity, such as file cabinets drawers, windows, doors and waste baskets, has the danger of overloading and confusing the scene by merging a limited virtual-world experience with that of real-world objects. Furthermore, the translation of abstract data values into real-world objects requires a limited visual metaphor with continuous consistency and a limited degree of interpretation freedom.

As the whole infoticle application takes place inside the virtual, three-dimensional world itself, the interface has to express a recognizable distinction between its information representation and its interface features. Therefore, the infoticle design categorizes *element functionality* by virtual *representation dimensionality*. Figure 3.39 compares the dimensionality of all elements used in a typical infoticle application.

- **1D Time Simulation.** The continuous simulation of time, the constant data streams and the interaction with time can be imagined to take place in a single dimension only.
- **2D Interface Elements.** All interface elements, including forces and filters, are represented by two-dimensional billboard icons that reside inside the three-dimensional scene. A heads-up legend display at the bottom of the screen informs users of the current state of the visualization system. This consistent two-dimensional design dramatically lowers the cognitive effort to recognize the spatial shapes of the tools.
- **3D Visualization.** All visualization elements that support users to comprehend the emergent patterns, such as trace ribbons and shapes, are inherently three-dimensional, provoking human cognitive capabilities to interpret the virtual world.
- **4D Infoticles.** Infoticles can be considered as mathematical objects that are placed inside three-dimensional space, and become animated over time.

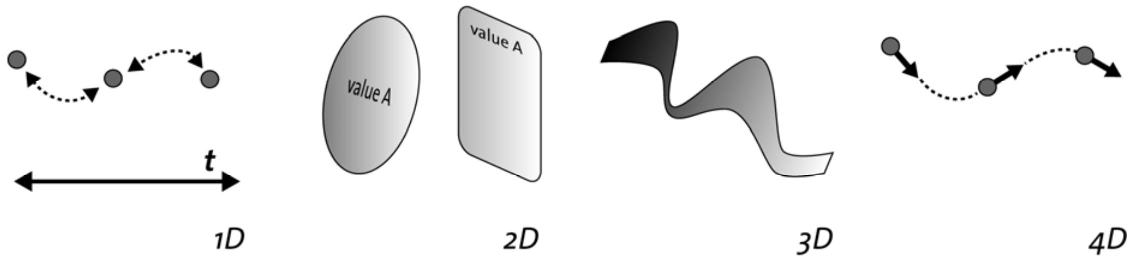


Figure 3.39. Representation dimensionality.

3.6.3. Interaction

The development of suitable interaction techniques is often neglected during virtual reality application creation. In contrast, the infoticle metaphor considers both as important and attempts to tightly merge visualization and interaction into a single system.

3.6.3.1. Input Device

Commonly accepted input devices such as mice and keyboards cannot be used within immersive virtual reality environments, as they require additional supporting furniture and would break the illusion of immersion. Instead, most virtual reality installations utilize a Six degrees Of Freedom (6-DOF) mouse, also called *wand*, as main input device (Figure 3.40). It is able to continuously measure both the exact spatial position and orientation within physical space. The wand device used at the blue-c portals is equipped with a joystick and three push buttons as additional means of input. The main disadvantage of such an input device is the strict requirement to constantly use fatiguing arm and hand movements. Furthermore, the input data retrieved by such a device are rarely precise, and often jigger rapidly within a dynamic precision range.

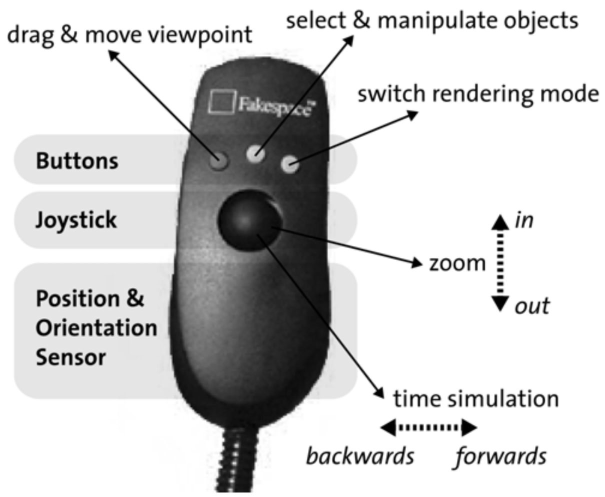


Figure 3.40. Wand.

Next to the tracker built inside the wand, immersive virtual reality installations employ an additional tracker on the user's head to determine the current physical viewpoint and viewing direction so that the exact perspective view of the computer-generated world can be exactly calculated and accurately rendered. This sensor is mostly used passively, and has no repercussions for the application functionality itself.

3.6.3.2. Navigation

One of the crucial characteristics of virtual reality is the ability to freely navigate inside the virtual space, to move around, to change position in the context of the spatial configuration and to explore the computer-generated representation from each arbitrary point of view. Navigation should allow a user to achieve a good spatial orientation that closely relates the current point of view with the information space context. This process has to be effective and intuitive, as the user navigates and orientates simultaneously in both the graphical world and the dataset structure.

Experiences gathered during the development of other virtual world applications have shown that users easily loose their spatial orientation when navigating within a three-dimensional world. As there is no physical orientation feedback within virtual worlds, the sense of left and right, and up and down becomes blurred from the moment the direction has been changed and the user's own physical orientation is kept constant. This phenomenon is especially problematic within information visualization spaces that translate data values directly into spatial coordinates. In fact, this very commonly used *graphical data mapping* technique requires the user to be always fully aware of the three fixed coordinate axes that form the main comparison basis with the spatial constellations. In addition, as virtual worlds have no physical borders, users can get lost or move too far away.

The infoticle application solves these space orientation issues by implementing a simple navigation metaphor that always points the viewing direction towards the visualization center. The navigation is similar to a *trackball* interface with *zoom* and *pan* control, as shown in Figure 3.41. By nature, it strictly limits the navigation possibilities of the user, who is able to focus on the central visualization and thus has less tendency to become spatially disoriented. The ever-changing infoticle movements produce a dynamic space that constantly transforms its formality. Consequently, specific static *orientation cues* are needed to let users create a distinguishable mental representation of the world. The infoticle system uses the static interface elements as spatial cues for virtual way finding.

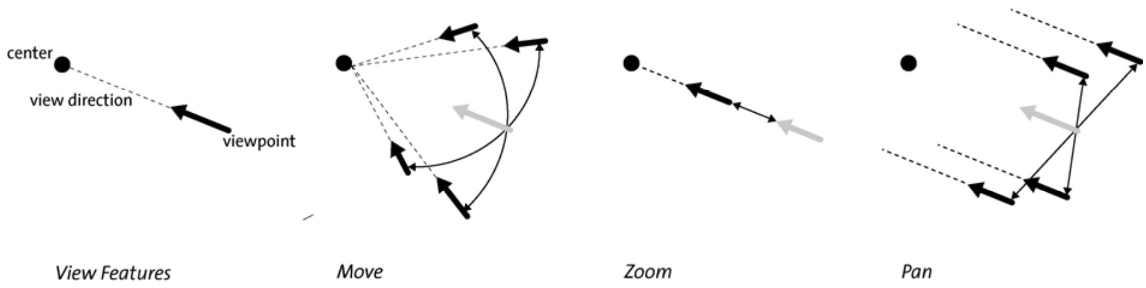


Figure 3.41. Trackball navigation.

3.6.3.3. Direct Manipulation

Shneiderman (1998) proposed the *direct manipulation* concept to interact with computer-generated applications. This approach is characterized by the following principles.

- **Continuous representation** of the objects and actions of interest, or continuous representation of the *potential for action*.
- **Physical actions** or selecting labeled buttons instead of a complex syntax.
- Rapid incremental and **reversible** operations whose effect on the object of interest is **immediately** visible.

This interaction technique converts physical user actions into the manipulation of graphical objects within the virtual scene. The impact on the object is immediately perceivable, as the user receives continuous visual feedback. Well implemented direct interaction tends to give users a certain sense of *control*, as they can instantaneously perceive whether their actions have the expected results, and possible counterproductive interaction can simply be undone. Furthermore, direct manipulation reduces *cognitive distance*, which is the distance between human thoughts and the physical requirements of the system in use.

The infoticle system employs this interaction technique extensively as literally all elements within the scene can be directly manipulated by users. In fact, the infoticle data visualization contains no user interaction which requires the comprehension of a complex interface syntax or a specific sequence of user actions. The main cognitive task needed by the data-driven particle metaphor is a global understanding of how the spatial characteristics of the tools must be modeled to reflect an explorative information representation. At a second level, these insights provoke users to influence the spatial infoticle constellations by manipulating the tools to reflect their personal data analysis expectations. Ultimately, an intuitive interface design encourages users to spend their time learning the task domain, and not the computer system.

3.6.3.4. Cursor

To avoid any cognitive overload of informational elements, the infoticle method utilizes a *flashlight* metaphor to present users with more detailed information about the visualized data object, as demonstrated in Figure 3.42. This method is conceptually similar to the *ray casting* metaphor and the *conic selection volume* described in (Liang and Green, 1994). In practice, this means that a virtual selection cone surrounding the infoticle cursor highlights all the infoticles within its perimeter. This technique focuses the user's attention on a relatively narrow field of view while still enabling a contextual overview of the surrounding representation. In practice, all infoticles within the flashlight-activated zone become visually complemented by a small, clickable circular icon, which facilitates selection with the spatial input device. In addition, a text label denoting the data object identifier is made visible and rotated towards the user.

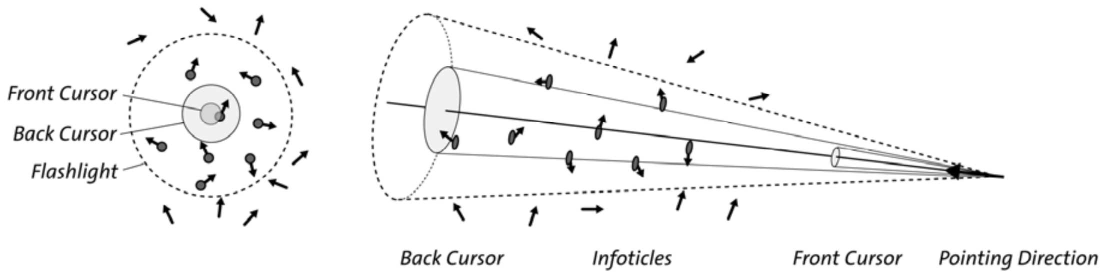


Figure 3.42. Flashlight interaction.

When an infoticle is preselected by the cursor, hovered by the front icon circle and clicked by the wand, all detailed data values of its corresponding data object are presented. The infoticle system does not show the actual multimedia document or database entry, as virtual reality is not efficient for showing media containing large text chunks, web pages or video documents. In effect, displaying real-world multimedia documents on a virtual reality display with such limited display resolution would utilize almost all available screen space for presenting intrinsically two-dimensional documents on a perceptually inefficient human scale. As a result, a user would not only find it difficult to comprehend the offered information, but also loose all contextual information about the three-dimensional information space surrounding that very object. Instead, it is preferred to store the data object selection within the system, which can be viewed and analyzed using more appropriate presentation media at a later point in time.

3.6.4. Collaboration

Virtual reality, and tele-immersion in particular, has proven to be an ideal technology to enhance collaboration when exploring and handling massive amounts of data (Reed, et al., 1997, Leigh, et al., 1999a, 1999b). Yet despite the many implementations of collaborative information visualization systems, so far user studies have been largely neglected (Mark, et al., 2003).

Many scenarios can be imagined that utilize the infoticle system capabilities for effective collaboration purposes. One of the initial goals of the infoticle method is to facilitate cooperation between remote users, ideally using the tele-immersive capabilities of the blue-c system (see Appendix A). Early experiments have shown the potential of bringing different experts together within a common infoticle visualization space. A shared infoticle space includes data-driven particles that intrinsically represent equal data objects on both sides, but present conceptually different data attributes to each participant, depending on their individual expertise (Figure 3.43).

At the same time, the unique blue-c capability to create real-time, three-dimensional representations of remote participants would be fully utilized, as physical three-dimensional gesturing plays an important role when interpreting and analyzing an infoticle visualization with more than one user: for instance, the blue-c technology would effectively represent the precise three-dimensional direction of a pointing gesture towards a single infoticle, which can be exactly spatially perceived by all participants within the shared information space. Clearly, this action would be ineffective when the participants are represented as three-dimensional avatars or on flat billboard surfaces.

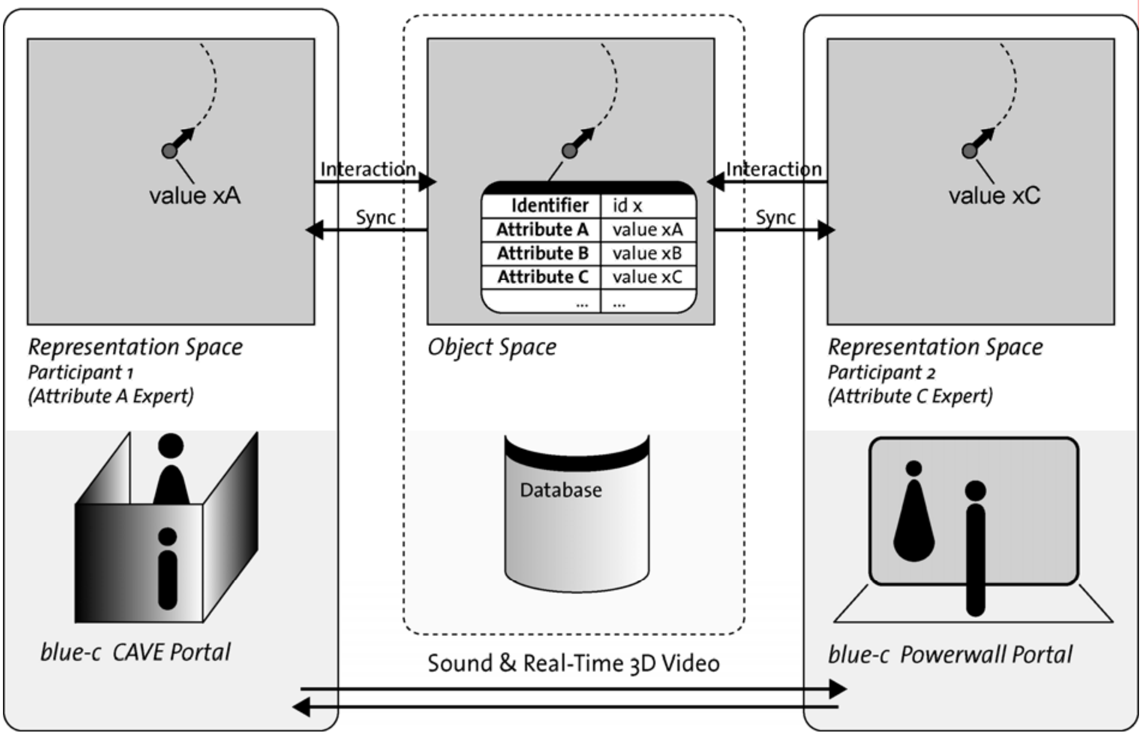


Figure 3.43. Infoticle collaboration.

However, porting the whole infoticle system to a practical implementation state that enables feasible remote collaboration is not trivial. In theory, all dynamic infoticle changes, including spatial transformation and data updates, need to be transmitted in real-time to all participating sides. In addition, all scene manipulations resulting from user interactions have to be adapted accordingly for all collaborating users.

As it is currently technically impossible to effectively communicate these huge quantities and frequencies of individual infoticle alterations, the most realistic implementation solution consists in transmitting solely the user actions, and to lock and synchronize the application time states in all participating applications. In effect, when all default initialization conditions are kept equal, the representation simulation will be exactly reproducible in time by all collaborating sides, and can only be changed by alterations invoked by the users themselves, which in turn are more efficiently transmittable and calculable.

3.7. Implementation

Creating a virtual reality experience requires both the design and programming of specially modeled virtual elements, interaction effects and sounds, suitable physics simulation algorithms, and intuitive user interfaces. As no authoring tools are currently available to support the handling and modeling of virtual simulations for immersive environments, many technical issues have to be solved on a highly conceptual level.

3.7.1. blue-c API

Developing virtual reality applications involves the combination of many different technologies at a relatively sophisticated core technical level, requiring the support and efforts of several experts from different disciplines. In practice, this means that application designers have to be aware of the requirements of both output and input devices, and have to use a low-level programming language to optimize the computing and rendering performance. Consequently, a specific *Application Protocol Interface (*API) is needed that frees the application designer from the relatively complex technical software and hardware issues of immersive virtual reality technology as much as possible.

An API can be imagined as a conceptual layer which resides between the graphical programming library and technical configurations on one side and the application code itself on the other side. It provides the application developer with easy access to the underlying virtual reality technology, and optimizes the computing performance for the available rendering hardware. In practice, an API offers several application features, supports the modeling and simulation of the virtual environment and presents the technical input device functionalities in a technically uncomplicated way. Currently, there are various *toolkits* available that provide support for developing virtual reality applications, including tracker libraries, scene graph protocols, networking tools and audio servers. However, the successful combination of these software libraries requires the understanding of many different interfaces and concepts that smoothly have to merge and cooperate. In fact, only a few approaches exist today that provide a more holistic, consistent and integrated toolkit that integrates all issues mentioned above.

The infoticle application development is built upon the *blue-c Application Programming Interface (blue-c API or bcAPI).* The blue-c API has been created in the blue-c project framework (see Appendix A) in order to aid the development of virtual reality applications that use blue-c hardware capabilities. The blue-c API is a software toolkit especially developed for media-rich, collaborative, immersive virtual reality applications. It is especially adapted to be employed on the blue-c hardware system and handles technical issues such as immersive projection, user navigation, two-dimensional video, audio, tracking, gesture recognition and the processing of interaction events generated by various input devices such as wands, mice, keyboards and trackers. The blue-c API tightly integrates the scene graph within its core functionalities to improve coherence, so that multimedia sources can be handled within the scene without compromising performance. Both the programming environment and the resulting code can be run on SGI IRIX as well as Linux operating systems. The API includes spline nodes, NURBS surfaces, animated scene graph objects and the like. For instance, the infoticle ribbon traces are developed by significantly adapting the blue-c API version of the Catmull-Rom spline nodes. The blue-c API even offers a simple particle engine, a functionality that was not used in this research to allow for full configuration and optimization of the infoticle data handling dependencies. The structure of the blue-c API is shown in Figure 3.44. In-depth technical explanations of the blue-c API functionalities and implementation issues can be found in (Naef, et al., 2003).

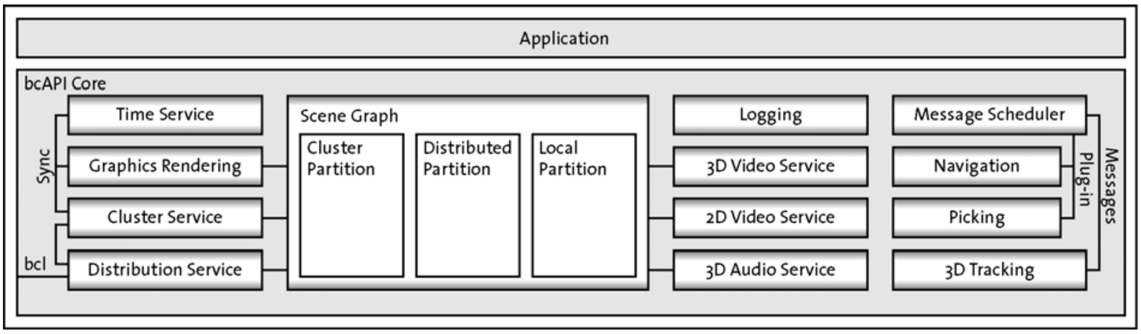


Figure 3.44. blue-c API structure (Naef, et al., 2003).

The blue-c API is built on top of the SGI OpenGL Performer API, which itself offers a scene graph and real-time rendering system (Rohlf and Helman, 1994), and uses parts of its process and shared memory management for process synchronization and communication. In practice, this means that the majority of the infoticle code is implemented using the SGI OpenGL Performer software library functionalities. In contrast, the blue-c API is primarily employed for virtual reality hardware configuration, spatialized sound simulation and interaction processing. All application code is written using the C++ programming language.

Most infoticle applications have been tested either on the blue-c hardware or on a Linux dual-processing machine. The blue-c rendering and computing hardware currently consists of a SGI Onyx 3200 machine equipped with two IR3 graphics pipes for the virtual reality theater, and a single dual-processor Linux PC for the powerwall. Two of the powerful features of the underlying OpenGL Performer library are the effective optimization of scene graph rendering, and the support of *parallel processing* with *shared memory management*, meaning that several application processes that share data can run simultaneously. As will be demonstrated in the next sections, this functionality plays an important role in guaranteeing a smooth and uninterrupted animation for applications such as the infoticle system, which typically require uninterrupted animation but meanwhile have to process large streams of data.

3.7.2. Forthcoming Database

Infoticle data processing relies on the *forthcoming database* middleware framework to query and deliver data on-demand. The forthcoming database concept supports the effective use of databases as interactive information sources (Mieusset, 2000). Typical data visualization applications offer users various sorts of personalized information, also called *awareness data*, which depend on events, interests and user interaction. Although such datasets are widely available and the demand for user-depending information is rising, only a few research approaches exist today that support application designers on the database system level.

In fact, instead of solely focusing on more traditional methods of iterative database queries and simple data retrievals, an alternative approach is needed to merge the aspects of visualization and database support. *Groupware* solutions focus on the interest of users, while database systems react solely upon dynamic alterations in the dataset. Both approaches typically neglect the visualization issues and are not as flexible as middleware systems that do not require specific proprietary systems and protocols.

In contrast, the forthcoming database concept represents a *middleware framework* that solves the lack of database support for visualizing awareness data. Such a conceptual prototype is independent of database management systems or application functionalities as it is layered in between the user application and the database itself (Figure 3.45). It offers transparent data access and is intrinsically *forthcoming* through various data modeling features and data preparation processes. In practice, this framework consists of several methods that application designers can exploit to structure and manage data depending on the interests and interactions of users. The specifications and architecture of this system are originally based on a collaboration tool implementation (Stouffs, et al., 1998).

The still ongoing forthcoming database research employs the infoticle visualization method as one of its applied case studies. In this context, the infoticle metaphor is treated as a generalized concept that utilizes extensively the forthcoming functionalities. The framework offers the infoticle application designer to dynamically organize data depending on user interests, such as the continuously updated active database timeframe. Furthermore, the forthcoming model is able to react upon expected information retrieval needs by *caching* data beforehand. Although the forthcoming database concept was integrated within the infoticle system as generally as possible, a close and intensive collaboration between the visualization and database developer was required. These teamwork efforts especially ensured consistent data communication and data handling between the infoticle application and the remote database, but also dealt with many issues regarding the required computing performance.

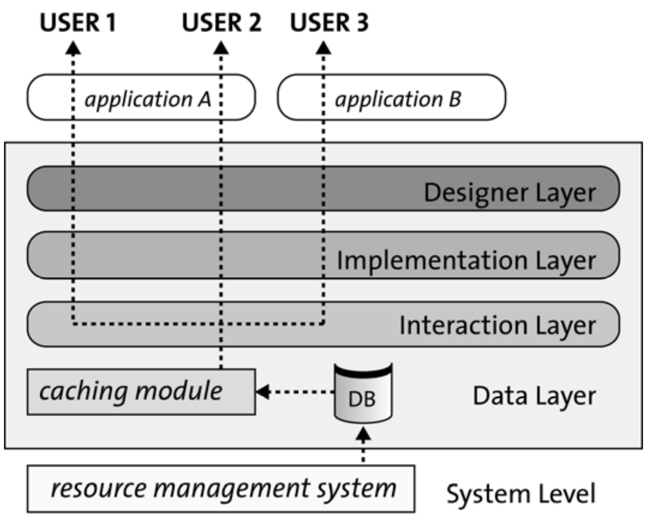


Figure 3.45. Forthcoming database structure.

Figure 3.45 demonstrates how the framework consists of several conceptual layers in which the data sources can be easily accessed and freely modeled by end-users as well as the application designer. The forthcoming database structure is a collection of *services* that the designer can employ during the application implementation without requiring expert knowledge about the database management system (System Level). The interaction layer acts as the fundamental base of the whole architecture assuring communication among users (i.e. end-user, designer) and data sources, and can also be used to connect different applications. The data layer has to be adapted by the application designer to implement the data handling processing required for the visualization simulation.

3.7.3. Data Processing

Two fundamentally different approaches can be followed when visualizing time-varying datasets. *Pre-computed animation* renders a dataset off-line, as the representation is calculated before it is shown. The resulting simulation only needs to render the pre-computed elements so that a predefined constant framerate can be guaranteed at all times. Consequently, the dynamic presentation can be displayed on low-level installations as all typical performance-expensive issues are overcome. However, such an approach results in animated movies in which a user cannot navigate nor interact, or in pre-recorded three-dimensional worlds where the interaction is limited to passive spatial navigation.

On-the-fly computed animation offers users the experience of smooth dynamics that react to real-time interaction. This concept requires large computing resources to be available, since frames must be continuously recalculated in response to changes in both incoming data streams and user interaction. On-the-fly computed animation is the only feasible method for information visualizations using virtual reality environments, as direct interaction is a crucial requirement of both fields. However, this approach has direct consequences for the software implementation of the proposed system.

3.7.3.1. Parallel Processing

The core of the infoticle metaphor relies on motion characteristics to convey changes in informational values. Therefore, rendered infoticle animations should never be interrupted by computing-expensive algorithms such as database calls or data look-up processes. Ideally, calculation-intensive processes are separated from those procedures that must not be disturbed.

Next to the infoticle updates and the continuous rendering processes, two different procedures take place in parallel: *interaction interpretation* and *data acquisition*. To make sure that these computing-expensive procedures do not collide, the application system supports *parallel processing* and *shared memory organization*. The infoticle application makes use of well-defined methods for structuring parallel programs by dividing the workload into a single *master* and several *slave* processes. This *master-slave* relationship can be imagined as a certain dependence in which a slave is capable of working, but is unaware of what should be done. Typically, a master knows what has to be executed, and instructs slaves on the work they must accomplish. This structuring mechanism is especially suitable to divide the workload in such a way that a specific process does not have to wait for a certain subtask to finish, but instead instructs a slave to carry out that work. For instance, in case of real-time information visualization, halting the animations while waiting for fresh data to arrive would thoroughly disturb the motion perception and dynamic pattern interpretation.

Figure 3.46 shows how each infoticle data object is updated within the application timeframe. Each infoticle checks whether a new data object is stored in the data cache for the actual database timeframe. In the meantime, the data processing timeframe awaits the signal for further database querying and data caching.

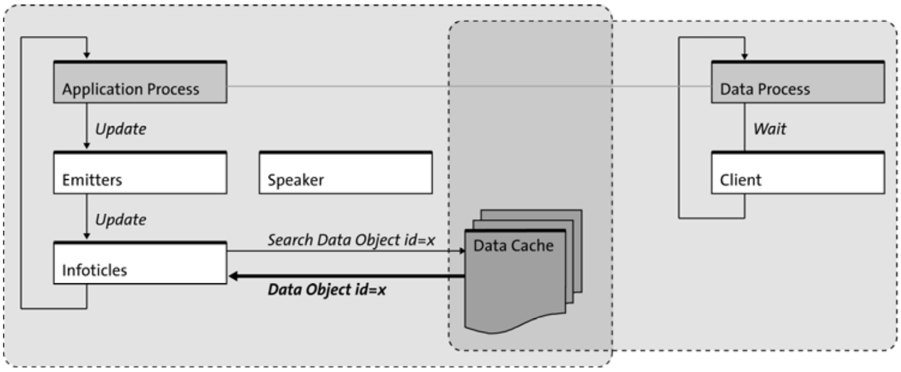


Figure 3.46. Data object update.

Figure 3.47 demonstrates the internal communication that takes place between the data acquisition process and the application process. In practice, the application process signals each speaker whenever the data caches of all emitters are cleared or the application timeframe has passed. Subsequently, each speaker informs the data process to query the database. Alternatively, the data process checks whether there is a need for new data when one or more of the data caches are empty. The returned data table is converted into separate data objects that are stored within an emitter-specific allocated data cache range. The application process takes care of synchronizing the emitters, so that all emitters become simultaneously updated. Each emitter data cache then contains all data entries that are necessary for updating the available data objects and corresponding infoticles to the actual application timeframe.

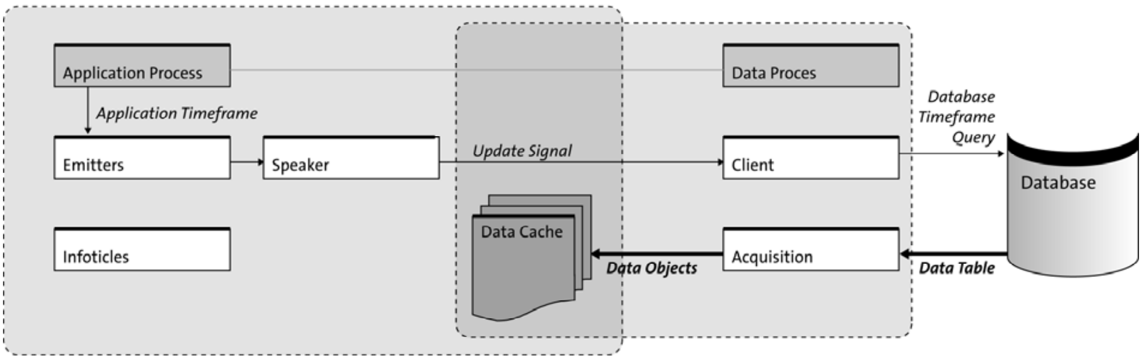


Figure 3.47. Parallel data object caching.

Notably, the processes shown in Figure 3.46 and Figure 3.47 are both executed in *parallel*, so that the visual effects of the infoticle animations are never affected.

3.7.3.2. Optimization

The infoticle method is geared to visualize relatively large, time-varying datasets. Consequently, the system must be able to update and animate thousands of infoticles without encountering performance issues that disturb an effective perception of the dynamic visualization patterns. Next to the obvious optimizations on the level of mathematics and software coding, the following conceptual approaches support the acceleration of the infoticle update process.

- **Particle Animation.** Different optimization approaches exist within the field of computer graphics to increase particle animation framerates. They include typical coding techniques such as efficient memory allocation and the division of particles in hierarchical trees according to their spatial coordinates. As a result, the rendering system is able to dramatically reduce the necessary rendering to those particles in spatial regions that are currently in view. The infoticle system does not use this approach, as the extreme wide view on the virtual world on the immersive displays would not necessarily result in large reductions of processed data.

- **Numerical Identifiers.** In practice, the application process needs to browse through all data objects within the data cache for each single data object update. Obviously, this look-up process becomes quite computing-intensive when performed for thousands of infoticles within the scene and a multitude of that quantity stored within the data cache for parallel sequential datasets. In practice, various if-then-else clauses check whether or not the data object identifier of the subjected infoticle and each single data object within the data cache are equal. This process is dramatically optimized when simple numerical values are used instead of textual real-world identifiers such as company names and documents titles.

- **Incremental Update.** A separate optimization process spreads the look-up process over the whole application timeframe, as shown in Figure 3.48. This method updates different, incremental infoticle groups during successive frames, instead of updating the whole infoticle collection at once within a single frame. This approach notably exploits the non-precise nature of dynamic particles, and the relatively slow perception of instantaneous motion events by the human eye.

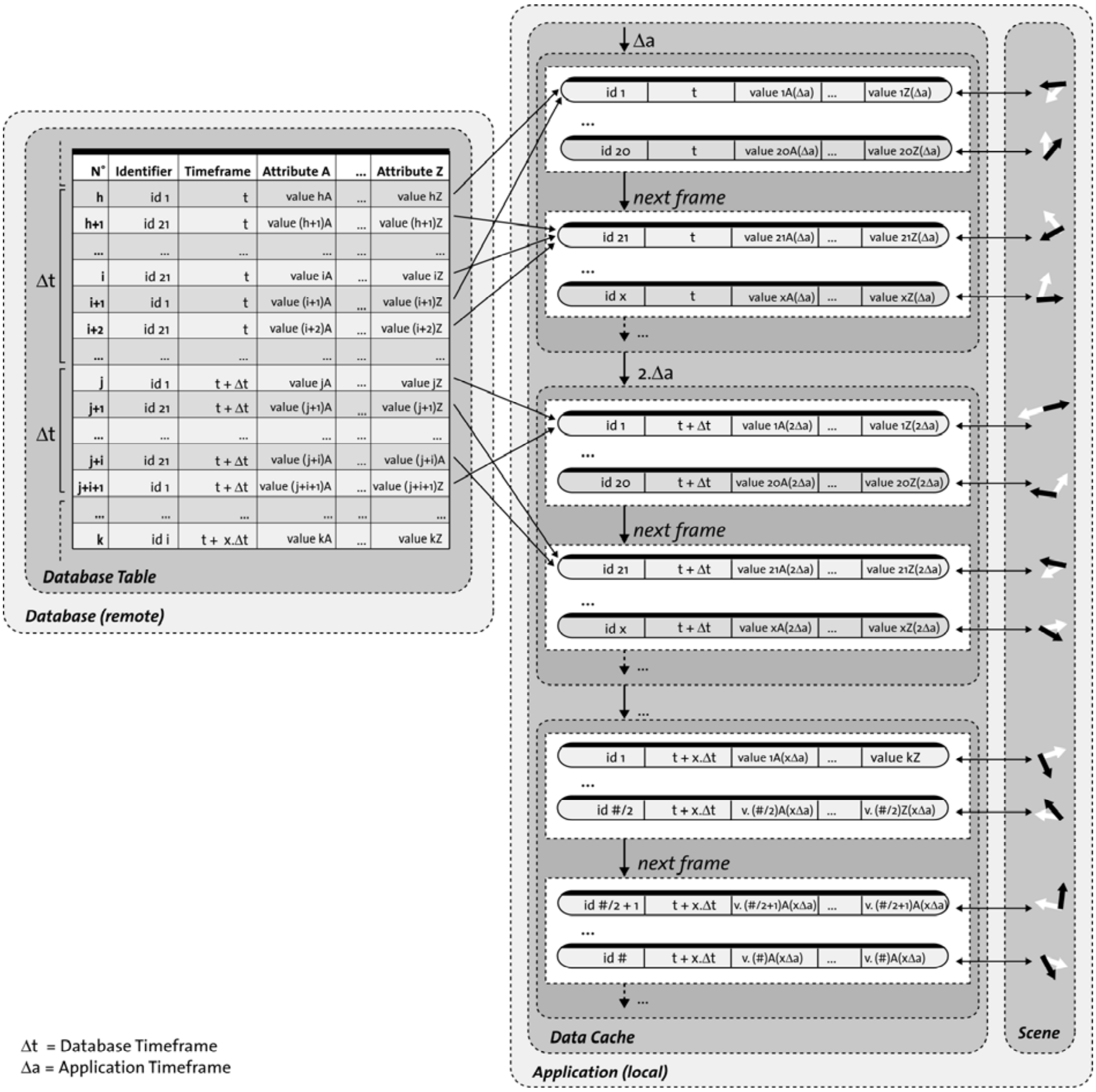


Figure 3.48. Incremental update optimization.

3.7.4. Simulation

The infoticle application exists of three processes that are executed at every frame: the infoticle world simulation, the processing of the interaction events, and the querying and caching of the dynamic data from the database. The *pseudocode* snippets within this section show the programming logic behind the infoticle metaphor. One should note that many algorithms are extremely simplified for reasons of clarity and understandability. For instance, elements such as infoticles, traces and shapes are not dynamically allocated in the computer memory during program execution as the necessary procedures would take up too much calculation time, and even might temporarily freeze the simulation. Instead, so-called *linked lists* are generated during the application initialization, and new and removed infoticle elements are continuously exchanged with these pre-allocated memory collections.

3.7.4.1. User Interaction

The infoticle system offers users a direct interaction mechanism. The following procedures are separated by the blue-c API from the main processes to guarantee an immediate evaluation of user-initialized interaction events.

– **Interaction Processing.** The user interaction process runs in parallel to the main application and continuously manages the signals generated by all input devices, such as keyboard, mouse and wand.

processUserInteraction (user_interaction_event)

```
case user_interaction_event of

  // SELECTION
  click:
    if an element within the scene is hit then
      highlight element
      engage dragging & dropping
      if element = infoticle then
        calculate trace out of infoticle history
        show trace
        show infoticle name in legend
      end if

  // CHANGE DIRECTION OF TIME SIMULATION
  time_direction_change:
    if time_direction_change = update then
      if time = update_time then time_direction = update
      else time_direction = transit
    else time_direction = time_direction_change
    show new direction in legend

  // ANIMATE FASTER SLOWER
  animation_speed_change:
    if time_direction = update then
       // IF CURRENT DIRECTION IS UPDATE THEN ALTER DATABASE TIMEFRAME
       if animation_speed_change > 0 then
         database_timeframe = database_timeframe – time_unit
       else database_timeframe = database_timeframe + time_unit
    else
      // ELSE ALTER APPLICATION TIMEFRAME
      if animation_speed_change > 0 then timeframe = timeframe – 1
      else timeframe = timeframe + 1

  // CURSOR MOVE
  cursor_position_or_orientation:
    update cursor cone position and orientation

  // NAVIGATE AROUND
  navigation:
    calculate new viewpoint coordinates & direction

  // CHANGE INFOTICLE REPRESENTATION
  rendering_mode:
    active_rendering_mode = rendering_mode

  // SOMETHING ELSE (MODE SWITCHES, DATA CACHING, )
  other:
    process user_interaction_event

end case
```

- **Viewpoint Redirection.** As a primary result of the user interaction evaluation, the viewpoint direction and position alter. These continuous navigational adaptations have important repercussions on all the two-dimensional elements within the scene, which all have to be immediately redirected towards the user's viewpoint. This process is especially computing-intensive when the infoticles are rendered as textures, as new directional vectors needs to be calculated for each single infoticle at each frame.

3.7.4.2. Application Simulation

The application simulation is the system's top procedure, executed once per frame.

- **World Update.** The world class initiates the whole scene, manages all elements in the scene and keeps track of various global variables, such as the actual time direction, the current viewpoint coordinates and direction, the time simulation speed, etc. The world update procedure invokes new data to be queried and cached when the application timeframe has passed, and updates all infoticles through querying the emitters.

```
updateWorld ()

// 'ACTIVE FRAME' COUNTER
set time to 0 (frames)

// LAST FRAME THAT DATA WAS RETRIEVED FROM CACHE
set update_time to 0 (frames)

// APPLICATION TIMEFRAME: WHEN NEW DATA NEEDS TO BE QUERIED (IN FRAMES)
set application_timeframe to 30 (frames)

// DATABASE TIMEFRAME
set database_timeframe to 24 (e.g. hours)

// SMALLEST POSSIBLE DATABASE DATA TIME GRANULARITY
set time_unit to 1 (e.g. hour)

// UPDATE EMITTERS
for each emitter
  do updateInfoticle
  if time_direction = update and mod(time,application_timeframe) = 0
  then store positions & data values of infoticles
end loop

// UPDATE TOOLS (FILTERS & FORCES & TEXT LABELS & TRACES)
for each tool
  update the tool according to current viewpoint
end loop

// UPDATE: CACHE DATA
if time_direction = update then
  if mod(time,application_timeframe) = 0 then cache new data
  updatetime = time = time + 1

// BACKWARDS: STOP AT START
else if time_direction = backwards then
  time = time - 1
  if time = 0 then time_direction = freeze

// TRANSIT: STOP AT END
else if time_direction = transit then
   time = time + 1
   if time = update_time then time_direction = freeze

end if
```

3.7.4.3. Emitter Simulation

The emitter is the only truly dynamic object within the world, and consists of a large infoticle collection. The emitter animation is dependent on the actual application time direction, and includes the following subtasks.

– **Infoticle Simulation.** By nature, infoticles have to be constantly animated, so that distinct, smooth and believable behaviors emerge. Technically, this means that at every frame, each infoticle has to evaluate all elements that influence its trajectory, and an updated directional vector has to be added to its current position. Subsequently, each infoticle must adapt its current speed and direction vector for every force in the virtual scene, and must check whether it passed through a filter or travels through the actual flashlight cursor.

– **Infoticle Data Update.** Every infoticle needs to check whether new data values have arrived for its corresponding data object for the currently active timeframe. Practically, each single infoticle has to browse through all data objects residing in the local data cache to search for one or possibly more corresponding data object(s). When new data has arrived, some of the infoticle attributes, such as speed, color and direction, have to be altered according to the data-driven behavior rules.

– **Particle Rendering.** This process is normally transferred to the graphics hardware, so that the available CPU power can be used for the previously mentioned tasks. In order to render and process particles efficiently, the particle collection should be well-structured. In practice, this means that rendering variables and visual settings should not continuously change or be different for every particle, and particle lists should retain a certain order. The thousands of particles are not programmed as atomic programmatic objects, as the scene graph would become too complex and too large for the necessary rendering and interaction tasks. Instead, a particle collection of an emitter consists of a single, huge graphical object with twice (line representation) or three times (texture representation) as many coordinates as available infoticles. The object's corners are drawn as lines or textured triangles at the specific positions of the infoticles. Especially the textured representation rapidly hits computing performance borders as each single texture polygon needs to be directed towards the user and the color values of dozens of overlapping, semi-transparent textures have to be calculated.

```
updateEmitter (time direction)

// ADD INFOTICLES TO SCENE
if number of infoticles < number of data objects in cache then
  for i < maximum number of new particles per frame
    add infoticle to scene with default attribute values
    assign free data object from cache to infoticle
  end loop
end if

// UPDATE INFOTICLE
for each infoticle of emitter

  // UPDATE THE INFOTICLE
  do updateInfoticle

  // ADD POSITIONS TO EMITTER COORDINATE ARRAY
  emittercoordinate[i] = position_previous
  emittercoordinate[i+1] = position_current

end loop

// RENDER EMITTER
if render mode = lines then
  render emitter by connecting pair with impair coordinates with a line
else if render mode = texture then
  render emitter by generating triangles with normal towards user
```

3.7.4.4. Infoticle Simulation

The infoticle coordinate calculation is probably the most calculation-expensive process of the system implementation, as these algorithms need to be executed for each single infoticle at each separate frame. The exact processing steps are dependent on the active time direction of the visualization simulation. In contrast to the backward and transit time directions that rely on the interpolation of the coordinates stored in the history lists, the update process requires the continuous checking whether new update data is available for the current database timeframe. Additionally, each infoticle needs to test all filters and forces within the scene for eventual spatial influences, or whether it is positioned within the flashlight cursor cone.

```
updateInfoticle (time_direction)

// TIME DIRECTION IS FORWARDS
if time_direction = update then

  // UPDATE DATA
  if new data object is in cache for current timeframe then
    delete old data object from cache
    assign new data object to infoticle
    store new data values into infoticle history list
    change certain infoticle attributes depending on behavior rules
  end if

  // FORCE INFLUENCE
  for each force
    if force data value = infoticle data object value then
```

```
        // CALCULATE DIRECTION: NEWTONIAN MECHANICS INFLUENCE
        v_new = v_old + f_force / m

      end if
    end loop

    // FILTER INFLUENCE
    for each filter
      if filter data = infoticle data object value and
        infoticle passes through filter then

        // CALCULATE NEW DIRECTION: BOUNCE
        calculate exact collision position
        calculate exact collision angle
        v_new = v_old + v_bounce

      end if
    end loop

    // STORE CURRENT POSITION IN INFOTICLE HISTORY LIST
    history_current = position_current

  // TIME DIRECTION IS BACKWARDS
  else if time_direction = backwards then

    // FIND PREVIOUS HISTORY POSITION WHEN TIMEFRAME IS OVER
    if mod(time,application_timeframe) = 0
      then history_current = previous in history list

    // INTERPOLATE DIRECTION BETWEEN CURRENT AND PREVIOUS POSITION
    v_new = (history_current – position_current) / mod(time, timeframe)

  // TIME DIRECTION IS TRANSIT
  else if time_direction = transit then

    // FIND NEXT HISTORY POSITION WHEN TIMEFRAME IS OVER
    if mod(time,application_timeframe) = 0
      then history_current = next in history list

    // CALCULATE NEW DIRECTION
    v_new =  -(history_current – position_current) / mod(time, timeframe)

  end if

  // ANIMATE INFOTICLE
  if time_direction != freeze then
      position_previous = position_current
      position_current = position_previous + v_new
  end if

  // CHECK IF INFOTICLE IS IN CURSOR CONE
  if angle(mouse->cursor, mouse->infoticle) < threshold and
     distance(mouse, infoticle) < threshold then

     // SHOW TEXT LABEL
     show a text label with data values of infoticle data object

  end if
```

3.7.4.5. Data Processing

The data processing procedures comprise a multitude of data conversions and look-up algorithms, and include the generation of database queries, networking and data caching. The following paragraph describes the triggering of the data acquisition processes and subsequent database and application timeframe updates.

- **Data Acquisition.** The data querying function is executed in another process that runs parallel to the main application. It waits until the main application signals that new data for the next timeframe needs to be cached. After the query has been sent to the database, the database requires a certain time span to calculate the resulting subset of data entries, to send the data through the network to the local machine and to store it inside the memory. As these algorithms are executed in a separate process, the performance of the animations and calculations of the main program remains unaffected.

```
cacheData ( time, timeframe )

// CURRENT STATE OF DATABASE TIMELINE
set frequency to 1

// RUN FOREVER (IN PARALLEL TO MAIN APPLICATION)
while main application is still running do

  // WAIT UNTIL LOCAL CACHE IS EMPTY OR APPLICATION TIMEFRAME IS OVER
  if local cache is empty or mod(time,application_timeframe) = 0 then

    // GO TO NEXT TIMEFRAME
    frequency = frequency + 1

    // CALCULATE NEXT DATABASE TIMEFRAME
    database_timeframe_start = start_time + frequency * database_timeframe
    database_timeframe_stop = start_time + (frequency+1)* database_timeframe

    // QUERY DATABASE AND STORE DATA
    query database for all data between start and stop
    wait for data to arrive
    order and organize data into data object format
    cache all data into array in local shared memory

  end if

end while
```

4. Application

Although information visualization is still a young and active academic field, only a relatively low amount of novel data mapping concepts have been invented and subsequently scientifically evaluated. In fact, most original information representation approaches originate from alternative, artistically inspired projects. Unfortunately, most of these experiments are overlooked by the scientific world as their usage potential and direct application is often underestimated. Unlike scientific visualization, the potential of information visualization in exploiting the qualities of virtual reality technology is still relatively unexplored, as reflected by the limited amount of related work mentioned in this thesis.

Restricting the invention of visualization methods to purely theoretical descriptions is often not satisfactory, as many practical aspects can only be discovered through empirical experimentation. In contrast, *demo-oriented* research enables a fluent transition from early conceptualization to hands-on development. In effect, building real-word working systems forms a key proof of concept to evaluate the success of a new idea, and implies a continuous assessment during the development process.

This chapter presents the various applications that have been implemented using the infoticle metaphor. By applying this visualization technique to fundamentally different datasets, these prototypes demonstrate the flexibility and versatility of the proposed data representation method. The prototype development process was characterized as top-down: the first application combines most infoticle features and system capabilities, after which the gained experiences narrowed the research focus and motivated the subsequent prototypes. The following applications illustrate various visual patterns that emerge from data-driven infoticle behavior and describe the corresponding informational interpretation. Furthermore, various user experiences gathered during the development process have been included. These insights will be used to support and document the formal infoticle metaphor analysis in Chapter 5.

The four different application prototypes are identified by names that describe their different conceptual perspectives on the infoticle metaphor.

- **Modeling.** The Modeling World prototype evaluates the possible user interaction mechanisms with the information representation, and analyzes the conceptual relationship between data exploration and spatial modeling within a virtual world.
- **Galaxy.** The Galaxy World demonstrates how dynamic update characteristics can be visualized using both externally and internally steered behavior patterns. The visualization system translates data update characteristics and time-dependent tendencies into cognitively interpretable dynamic reactions and static shapes.
- **Electron.** The Electron World mainly functions as a dynamic presentation medium capable of illustrating time- and dimension-varying data changes. This visualization scenario employs time animation interaction and force set switching to effectively represent multi-dimensional data.
- **Boid.** The Boid World uses self-organizing principles to generate spatial behaviors out of rapidly altering datasets. The generation of shapes enhances the interpretation of the emerging visual patterns and spatial clusters.

4.1. Design

Demo-oriented research, and application prototyping in particular, is often influenced by many contextual factors. The following sections list the followed sequence of development stages and describe the general motivations that drove the application design.

4.1.1. Concern

The main goal of implementing fundamentally different applications is to demonstrate the versatility of the infoticle visualization metaphor. By consequence, a thorough visualization method evaluation was preferred as opposed to a detailed dataset analysis. Therefore, the following considerations need to be taken into account when evaluating these application prototypes.

- **Dataset.** It was chosen to limit the required information technology complexity to handle the datasets efficiently. In practice, this means that sophisticated data mining tools or professional databases have to be avoided. Instead, easily accessible datasets are preferred that require relatively simple software tools to be effectively converted and queried. One should note, however, that the relatively simplicity of the datasets chosen might be exploited to dispute the true intrinsic values of the resulting data patterns.
- **Presentation Medium.** The use of relatively sophisticated virtual reality equipment for visualizing these kinds of datasets might seem disproportionate. Yet, these datasets are implemented primarily as manageable sources for early experimentation, with the goal to explore the effectiveness of the infoticle metaphor and detect its potential features for information display in immersive environments.
- **Value.** Instead of gathering true insights out of valuable and unexplored data, these applications are focused on demonstrating the potential of the infoticle method. In fact, because of its absolute novelty, the infoticle method's true usability and validity is

difficult to determine without an initial analytical evaluation, a task which this thesis attempts to accomplish.

- **Experience.** Most of the prototype features might be difficult to appreciate without experiencing them first in person. For many observers, the visualization concepts seem rather unusual because few real-world analogies exist. However, when one becomes accustomed to the features of immersive virtual reality and learns the navigation and interaction controls, the informational journey becomes quite interesting and engaging.
- **Image.** The application images attempt to convey the achieved visual quality of the information representation. However, one should note that the third dimension, i.e. depth, becomes merged unto the two-dimensional picture, increasing the perceived picture complexity considerably. Next to the unfortunate fact that picturing a three-dimensional world hides its spatial simplicity, it cannot represent the nature of dynamics either. Consequently, different motion features will be presented by showing sequences of static images, or by framing the spatial artifacts that trace the dynamic nature of the time-dependent events.

4.1.2. Development

Most of the prototype applications adopted a *user-centered* design approach, as they address certain requirements that were expressed by prospective users. A process of negotiation determined the following contextual parameters, which in turn influenced the outcomes of the different application developments.

- **Context.** In general, information visualizations are capable to be used in different conceptual contexts. The infoticle applications have been particularly designed and subsequently employed for the following purposes.
  - **Data Exploration.** The user utilizes the visualization simulation mainly for gaining a general understanding of the dataset characteristics in an iterative and explorative way. Typically, the user is a non-expert in the field of the subjected data attributes, and wants to get accustomed to the specialized time-varying tendencies and data value characteristics. In this context, the infoticle method is especially useful for discovering the typologies of individual and global data updates.
  - **Data Model Comparison.** The user analyzes in detail the spatial information representation and is an expert in the specific dataset phenomenology. The experience gathered is used as navigational knowledge to discover detailed tendencies within the data. The user focuses on specific data patterns that fit certain expectations, out of which an interpretation model emerges. Accordingly, the infoticle method supports such comparison process by facilitating analysis on different contextual levels of detail.
  - **Information Presentation.** The resulting visualization is used for presenting specific time-varying trends and changes within the dataset. Mostly, these tendencies are known, but are difficult to understand for non-experts by traditional visualization means. Therefore, the simulation allows for such audience to understand these phenomena by distinguishing emerging visual patterns. Interaction features are limited to simulation alterations such as time direction shifting and system mode changes.

- **Presentation Medium.** All application prototypes were originally developed to be used on an immersive virtual reality system, although most of the implementation was accomplished employing normal desktop hardware. As will be demonstrated, the insights gathered during the first prototype evaluation directed the subsequent applications to a more abstract level, at which specific dynamic data mapping concepts could be observed in more detail. During this development process, the importance of the presentation medium received less attention, but was never neglected.
- **User Community.** The typical user community for these prototypes was quite small and generally consisted of only a handful of people who were familiar with the datasets. This small target audience enabled the application designer to fine-tune the visualization at an expert level, but also resulted in a system functionality that might be more difficult to work with for inexperienced users. In effect, many information visualizations are typically utilized by a limited amount of people only, as most datasets are highly specific in nature and need a certain degree of specialization to be fully comprehended. Furthermore, the necessity to experience the infoticle system within a virtual reality environment limits considerably the possible audience size that can effectively employ it in a real-world context.

4.1.3. Process

In practice, the invention and subsequent development of an information visualization application goes beyond coding application functionalities, and includes a set of successive steps that starts from acquiring raw datasets and reaches to improving the data mapping and interaction algorithms. The next sequence of development stages has been followed during the various application implementations.

- **Scenario Development.** This phase urges the dataset owners to consider the intrinsic informational values of the acquired dataset. During the scenario evaluation, an attempt is made to predict the possible data patterns that might be hidden within the data and might interest the targeted audience. Several interviews are conducted with potential users and data owners to discover their expectations of the visualization application. These considerations are required to decide whether the infoticle metaphor and the virtual reality presentation medium are well-suited for visualizing the specific dataset. Therefore, infoticle capabilities, dataset characteristics and user goals are compared and evaluated. This phase typically results in a detailed description of the *infoticle visualization scenario*, which includes an explanation of the offered application features and the desired data mapping algorithms.
- **Data Acquisition.** This phase deals with the compilation of the accumulated data in a single format with fixed attributes decipherable by the infoticle system. In the simplest case, all data entries are collected from an electronic file and dumped directly in a database. More complex scenarios actually require the retrieval of raw data from various databases, documents, online resources or real-time data streams, often requiring a time-consuming conversion process. As the acquisition process depends on data source characteristics, it typically needs to be re-implemented for each dataset type.
- **Data Filtering.** Many datasets are incomplete, contain inaccurate or irrelevant data entries or include data values that cannot be effectively managed by the parallel data-handling processes. The filtering phase transforms or deletes these data values, hereby guaranteeing the integrity and consistency of the whole dataset between certain predefined accuracy thresholds. This process ensures that the infoticle application is

able to process and represent all stored data values, so that no unexpected occurrences break the visualization metaphor. In practice, this phase typically verifies whether files contain invalid data entries with delimitating characters used by the infoticle communication protocol or have no time identification stamps.

– **Data Abstraction.** The raw dataset is converted into a higher level of abstraction by a process that links equal data values and orders them in time. For instance, many datasets contain hierarchical entries that need to be categorized in specific arrangements, or include different entries for a conceptually identical data object. These values thus possess meaningful singular relationships that must be stored within the database. After the different files have been merged inside a single timeline, all data entries are cross-matched to detect data value analogies, so that the change of identical, reoccurring data objects can be recognized and stored.

– **Data Conversion.** The abstracted datasets need to be transformed into an electronic format that can be processed by the infoticle visualization application. This implementation level includes the design of database tables specially adapted to the infoticle protocol. An infoticle table generally contains entries that discriminate each unique, reoccurring data object and the values of that data object in time. This phase also maps categories of textual data onto corresponding numerical values, which compresses the data size considerably and enhances drastically the conditional processing during the infoticle simulation.

– **Metaphor Adaptation.** Because the visualization metaphor has been adapted to the dataset characteristics in the first implementation step, the data mapping algorithms need to be changed accordingly. These alterations usually take place on the level of the structural layer (see Section 3.5.1. Infoticle – Simulation – Data Flow). As the tool variables, infoticle attributes and self-organizing behavior algorithms are highly interdependent, several fine-tuning adaptations are needed to ensure the robustness and effectiveness of the emerging visualization patterns.

– **Application Reconfiguration.** Often, small application level alterations need to be implemented for each infoticle visualization scenario, although these operations are usually limited to defining new spatial constellations of infoticle tool objects. However, these operations might also include adaptations on the interface level to accommodate new interaction paradigms, and data processing optimizations that depend on the dataset size and typology.

Ultimately, these different implementation levels might serve as the foundations of an *infoticle visualization framework* that enables visualization designers to incorporate the infoticle system within independent applications. Unlike the current application state, a *framework* is able to adapt to dataset requirements and end-user expectations in an automatic and user-friendly way. A first step in this direction would include a configuration tool that can be used by a application designer before or even during the visualization presentation on a remote machine. Such an interface should facilitate real-time adaptation of various infoticle system variables, hereby avoiding hard-coded alterations within the software. Furthermore, a future infoticle framework could offer similar information processing applications a versatile programming interface (API) to incorporate the infoticle metaphor within different visualization and dataset contexts.

4.1.4. Dynamic Data Characteristics

As the various applications will demonstrate, the infoticle method is capable to visualize various data dimensions that are mostly *indirectly* stored within time-varying datasets. Accordingly, infoticle behavior generation is able to represent the following dynamic data characteristics.

– **Data Value.** The exact set of numerical or nominal values, labeled by the data attributes of a data object.

– **Data Update.** The exact occurrence of data values being changed to the values of the next database timeframe, driven by the rhythm of the application timeframe.

– **Data Value Change / Data Value Update.** The relative change of data object values after a data update. The alteration of numerical data values from one timeframe to another can be simply evaluated. The relative change of nominal data values are expressed by comparing the occurring frequencies within the active database timeframe. Conclusively, data value changes can be compared with the rest of the dataset for that database timeframe and be characterized e.g. as significant, relatively large, etc.

– **Data Value History.** A list with the past time-dependent data values of a data object. By comparing the data value changes within a data value history, the evolutionary characteristics of that object can be discovered and compared to others.

– **Data Update Frequency.** The frequency with which a data object is subjected to alterations during a single application timeframe. For singular sequential datasets, the data update frequency is always one or zero, for parallel sequential datasets it is unknown beforehand.

– **Data Update History.** A conceptual list containing the sequence of data updates. Such list is not necessarily saved in a computer's memory, but might be represented through visual artifacts, consisting of the passed data update frequencies and relative data value changes. Typically, the infoticle trace ribbon represents the data update history of its corresponding data object.

Notably, the exact data values, relative data value changes and data update frequencies of a data object are the three main determinants of infoticle behavior.

4.2. Modeling

The *Modeling World* allows users to browse various types of time-varying datasets and explore the concurrent dynamic characteristics. This visualization scenario mainly focuses on the iterative aspects of exploration by facilitating the *modeling* of a user-specified data querying world within the three-dimensional information representation space. In effect, the modeling visualization patterns are not mainly generated by local interactions of behavior rules, but instead by user-defined tool influences within the virtual world itself. Therefore, users need to directly adjust the spatial characteristics of these tools to reflect their personal data analysis interests.

This information modeling concept demonstrates how the infoticle system is able to provoke a user's spatial reasoning when investigating the hidden patterns of an abstract dataset. In effect, a well-considered constellation of tools and sources can be rightfully imagined as a three-dimensional *diagrammatic representation* that spatially tracks the user-specific data exploration process. Typically, the positions and orientations of the data querying elements are designed following a data exploration process (see Section 2.2.2. Background – Information Visualization – Exploratory Data Analysis), consisting of iterative trial-and-error experiments. In practice, the tools are spatially rearranged through a process of direct manipulation, as they continuously influence the infoticle representation in real time. These modeling principles are demonstrated trough an application prototype that visualizes a subset of ETH-Zurich financial budget data.

4.2.1. Goal

The Modeling World acts as the first case study of the infoticle visualization method. Consequently, this application is specifically created to evaluate the perceptibility of the visual data patterns and the effectiveness of the interaction mechanisms. One should note that many functional aspects of this experimental world are fundamentally different from the general infoticle method described in Chapter 3, as most behavioral details and data processing mechanisms have only been specified after the assessment of the Modeling World outcome. Consequently, the primary goals of the Modeling World are related to metaphor experimentation and basic methodology evaluation.

- **Dataset.** The main objective of the Modeling World is to demonstrate the potential of the infoticle method to a wide audience. Therefore, the dataset ideally needs to be generally understandable and publicly available. In fact, the dataset contains the yearly financial budgets of university departments and the numbers of students within these departments, facilitating a simple time-varying comparison between the amounts of students and the resources acquired by each academic department. Both these dataset types show a low data complexity, and share the different university departments as a common data dimension. In addition, both the data quantity and dimensionality are relatively small, so that data updates can be simulated locally instead of being fetched and processed from a remote database.
- **Infoticle Method.** The Modeling World uses the multiplied infoticle initialization to interpret numerical data values, which facilitates a qualitative rather than a quantitative comparison of the resulting infoticle clusters. The data mapping method itself is reduced to a minimum and is implemented directly from relative data value change to infoticle attribute alteration (without any behavior rules), to gather first impressions about the true 'data-driven' effects. This scenario thus functions as an experimental basis for determining more detailed infoticle behavior rules in future prototypes.
- **Interaction.** Because of its focus on immersive modeling, the Modeling World offers intuitive interaction mechanisms that overcome the practical limitations of virtual reality input devices in order to enable an effective spatial design and thus data exploration process. The modeling procedures are kept to a minimum and require no extra cognitive overhead. However, most fundamental usability issues related to immersive virtual reality technology that allow for the intuitive modeling of three-dimensional worlds are still largely unknown.

4.2.2. Implementation

Because of the novelty of data-driven particles, their effectiveness has to be thoroughly assessed before more critical data processing issues can be handled. Most insights that led to the infoticle core mechanisms were gathered during the Modeling World evaluation, so the following features might differ from the methodology explained in Chapter 3.

- **Dataset.** As the first infoticle prototype, the implementation effort was reduced to the core metaphor mechanisms. Therefore, it was chosen not to retrieve the dataset from a remote database, but to hard-code the data objects locally. This radical feature reduction allows for a metaphor evaluation freed from complex data processing algorithms and computing performance constraints. The dataset is condensed to three numerical data dimensions and contains only three database timeframes.
- **Interaction.** The main interaction features facilitate the alteration of the tool positions plus the spatial directions of filters and sources. Users are able to select and subsequently pull, push, pan and rotate all elements using the wand input device.
- **Infoticle.** The Modeling World employs no infoticle behavior rules in order to observe the effectiveness of the visual patterns when the spatial influences are solely determined by the laws of Newtonian mechanics. Furthermore, the infoticle attributes are complemented by a lifespan value, after which the according particle faints out and is removed from the scene. Because of the limited data processing overhead, the Modeling World is able to effectively handle more than 12.000 infoticles simultaneously.
- **Data Update.** Instead of uniquely symbolizing a specific reoccurring data object, a Modeling World infoticle represents a single multiplied data object that is limited in time. At the rhythm of the application timeframe updates, old infoticles are removed from the scene and new ones stream out of the infoticle emitter source. Consequently, the data value evolution can be observed by repeating the same successive timeframes, as the infoticle system starts over the entire timeline whenever the complete dataset has been simulated. This continuous sequential repetition of database timeframes enables users to observe how infoticles react to specific spatial filter, force and source constellations. Subsequently, users have the chance to adapt the spatial attributes of the information querying virtual objects according to these empirical findings in an iterative and direct way.
- **Source.** Fresh infoticles enter the three-dimensional representation at the infoticle emitter source position. The infoticles stream out of the emitter source in a time-ordered way, so that the oldest data entries within the database timeframe enter the scene first, followed by the successive time entries in the dataset within a single database timeframe. The source emits new infoticles until all data objects within the database timeframe are present in the scene. Infoticles of the next database timeframe are added when the application timeframe has ended. Users are able to alter the exact location and direction of the emitter source as its spatial context plays a very important role in the data-querying spatial tool constellation.
- **Density Map.** The textured representation mode facilitates the blending of clusters, resulting in a spatial map with regions of gradual color intensities that represent the relative quantities of infoticles parallel to the view direction.

– **Tools.** The Modeling World representation is governed by the local and real-time interaction of all infoticles with the tools. Consequently, individual infoticle behavior plays a less important role in the interpretation of the data representation, as the data-dependent conditions of the tools generally are valid for large quantities of data objects.

– **Interface.** The interface is limited to the tools within the world, which are represented by simple geometric shapes. Because the Modeling World expresses a horizontal directionality, navigation is limited to the classic fly instead of the trackball metaphor.

4.2.3. Result

Figure 4.2 shows the clustering influence of a single force, attracting all infoticles with an equal data value in a *jovicentric* trajectory, and leaving the others unaffected.

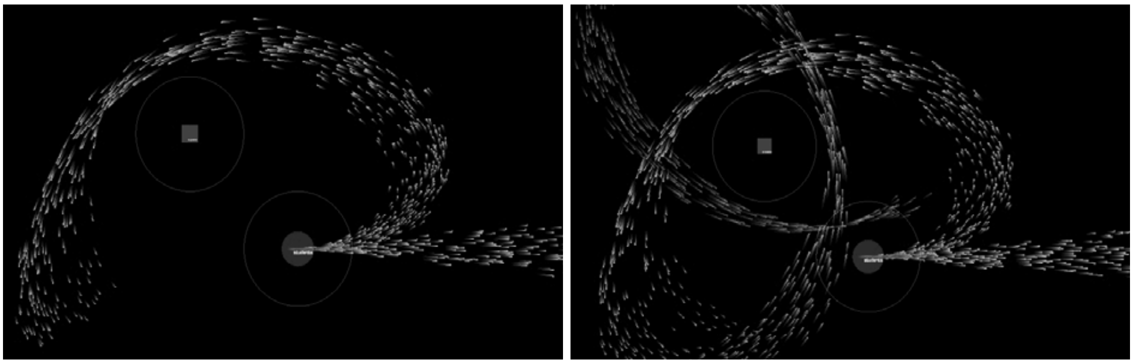


Figure 4.1 Infoticle force.

Furthermore, Figure 4.2 illustrates how a filtering surface physically bounces back infoticles with different data values, and separates them from the others.

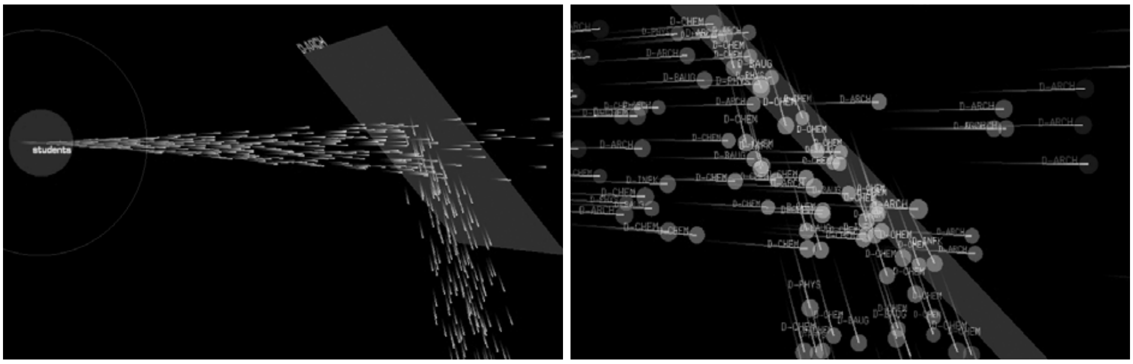


Figure 4.2. Infoticle filter.

Figure 4.3 shows the basic elements of a Modeling World scene and the visual outcome of a simple set of sources, forces and a single filter. All infoticles represent either students or money, and stream out the corresponding sources in a time-ordered way. These infoticle flows become influenced by the tools within the scene, bounce against filters or cluster around forces. User-defined manipulations have immediate implications on the infoticle stream evolution. In particular, Figure 4.3 demonstrates how two infoticle flows that stream out of sources (depicted by circles and representing money and students) are affected by two forces (shown as square icons and symbolizing the departments of chemistry and architecture) and a single filter surface (signifying the department of architecture). Architectural infoticles cluster around the force, chemistry-bound infoticles bounce back from the filter, whereas all other infoticles are unaffected by this spatial constellation of tools.

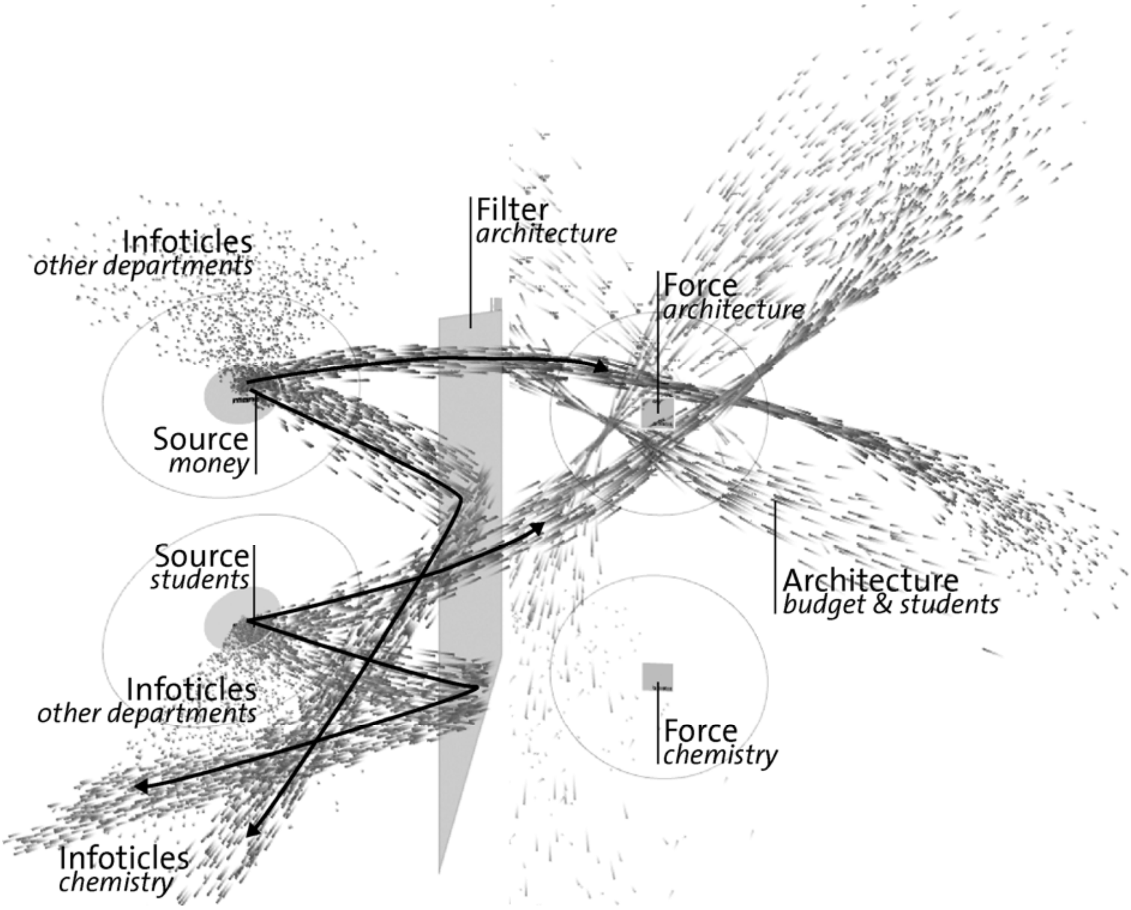


Figure 4.3. Modeling World scene.

Because of the quantity, global density and smooth dynamic nature of the data-driven particles, the Modeling World conveys highly immersive characteristics, even when shown on traditional displays without stereoscopic capability. This phenomenon supports the feeling of presence and spatial orientation, so that cognitive tasks are limited to the comprehension of the representation. Figure 4.4 shows the size and visual impression of the Modeling World in comparison to users. One can perceive the large physical display size and the infoticle text labels within the cursor cone.

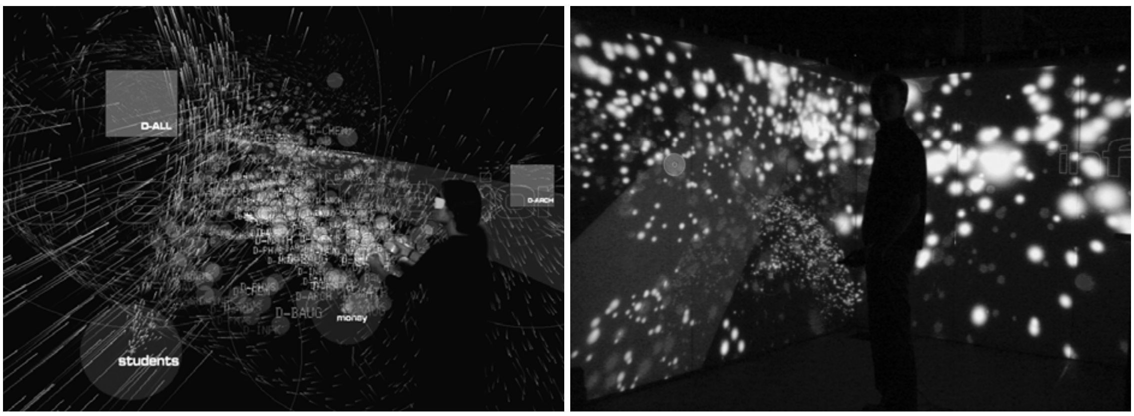


Figure 4.4. Modeling World immersion.

Figure 4.5 demonstrates how detailed data values become visible when infoticles are selected by users.

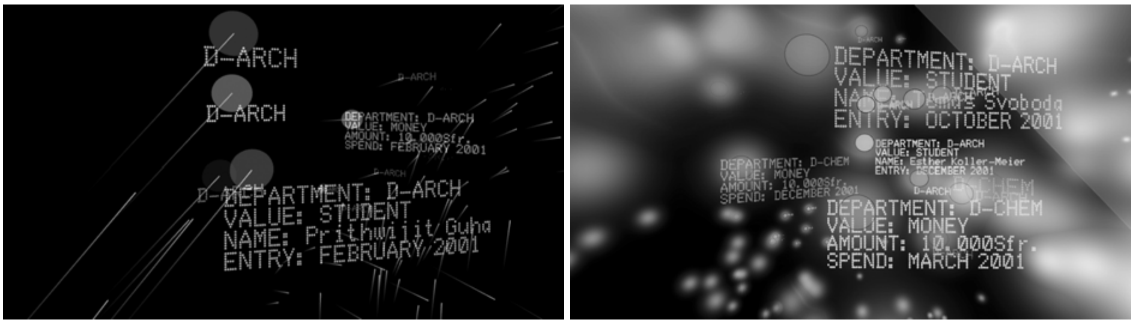


Figure 4.5. Infoticle information label.

Figure 4.6 illustrates how the density and relative size of infoticle clusters are more efficiently perceivable when rendered in the textured representation mode. Instead of simple lines, infoticles are displayed as semi-transparent spots that spatially overlap. Due to its high visual quality and the high amount of blending textures, this mode results in a very slow framerate that makes effective interaction almost impossible. Notably, although these figures might seem visually complex, they look relatively empty when perceived in a stereoscopic mode.

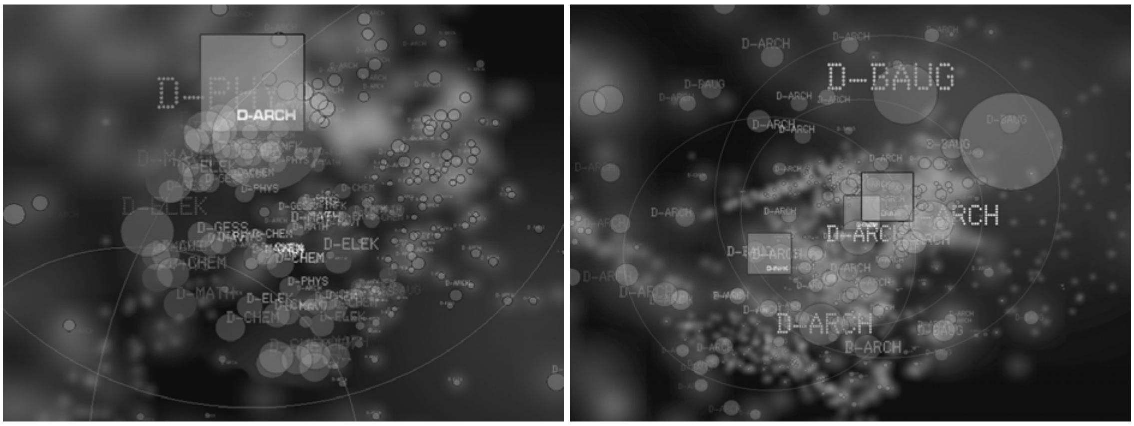


Figure 4.6. Textured representation mode.

After the application initialization, all elements in the scene are positioned at default locations, which the user then can change at will. Figure 4.7 shows how the different relative positions and orientations of the sources, forces and filter within the world produce fundamentally different infoticle collections with equal data values. One should note that these representations are normally perceived on a huge scale, with filters appearing as large as physical doors.

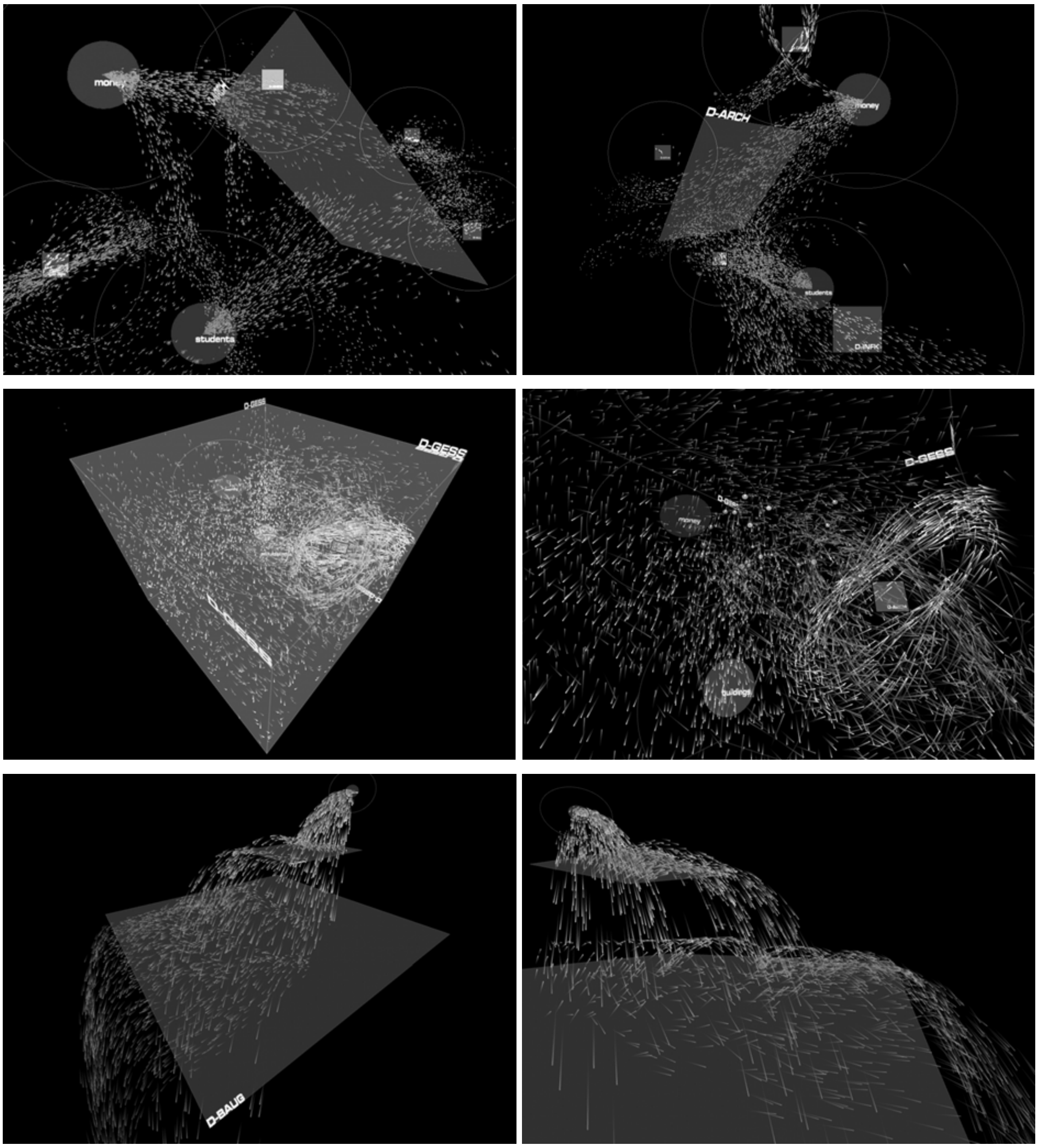


Figure 4.7. Tool modeling.

4.2.4. Evaluation

The following observations have been made during the Modeling World demonstrations.

– **Influence.** As the Modeling World is not governed by local behavior rules and is solely based on the natural effects of Newtonian mechanics, its visual patterns are relatively chaotic. Some infoticles crash into the center of the forces or leave initially stable orbits due to user-initiated events. Infoticles at larger distances from the forces often remain unaffected, as the resulting attraction strength decreases with increasing distance. Furthermore, infoticles moving at relatively high speeds are not significantly influenced by force attractions. Consequently, extra behavior rules are necessary that stabilize these trajectory changes and guarantee effective force influence and thus data-dependent behavior regardless of infoticle distance or speed.

- **Reconfiguration.** The data conditions of the tools cannot be reconfigured at runtime. In fact, the Modeling World principle is based upon the hypothesis that the expected queries required during the data exploration process are considered beforehand, so that the necessary tools can be made available to the user. Alternatively, the infoticle scene could literally offer a whole *library* of possible tools representing various possible data conditions, out of which the user is able to choose and subsequently model the personalized information exploration environment.
- **Data Update.** Users have expressed difficulties with distinguishing the continuous repetition of database timeframe streams and are often unaware of the actual active timeframe. Consequently, some sort of visual time discriminator is needed that, for instance, enters the scene between the different time-based infoticle batches to spatially divide the different groups of timestamps. Alternatively, a legend console with textual information could provide the user with the active timeframe and infoticle system mode.
- **Time.** Users have no capability to manipulate the direction or speed of the time simulation, and thus were unable to replay sequences or to zoom in on sudden events. Instead, the Modeling World offered a continuous timeline repetition, so that users could reconsider their actions in an iterative and repetitive way.
- **Interaction.** Although users appreciated the direct interaction, a considerable amount of time and effort is required to model the environment as a useable information-filtering constellation. Preferred spatial arrangements can be *dumped* and reinitialized on later occasions, although this process happens at the source code level. In effect, some sort of user-friendly constellation storing feature is needed when user interaction plays such a predominant role in the data pattern generation.
- **Patterns.** Due to the absence of different infoticle motion typologies, it is difficult to identify differences in dynamic infoticle behaviors. Instead, most infoticles move smoothly and similarly, and thus leave no discriminating trace with informational value. Consequently, clusters and infoticle-force proximities denote some degree of similarity, but do not fully exploit the potential of motion for information display.
- **Usability.** Users are compelled to interact within the infoticle world, although had difficulties to achieve detailed modeling due to the imprecise input devices employed. As the exact positions and direction of the tools had important repercussions on the data-querying representation, users wanted to place them very accurately, a task that challenges the current constraints of immersive virtual reality input technology.
- **Interface.** The overall interface design engages users to manipulate the virtual world, and proves the potential of virtual reality environments for interactive information visualization purposes.
- **Metaphor.** This early infoticle metaphor experiment demonstrates the potential of immersive information visualization methods that do not employ traditional static spatial mapping algorithms to represent data attributes. Instead of estimating Cartesian coordinates, relative distances and quantitative clusters show trends within the data.

4.2.5. Conclusion

The Modeling World is fundamentally different from the other infoticle scenarios in that it generates the visual infoticle patterns solely through a continuous *process of user interaction*, and not by local behavior rules or self-organizing principles. In effect, users are offered thousands of flowing data-representing particles that can be influenced in real time, but show no initial coherence. They are constantly exposed to a three-dimensional world filled with data objects, and have to make their own unique conceptual model to be able to comprehend the internal structural relationships.

The *information modeling* concept shifts cluster creation and data comprehension from automatic data mapping algorithms figuratively into the hands of the creative user. Tremendous amounts of data literally float in space, and are sorted and clustered by user interactions until some sorts of visual patterns are generated and subsequently discovered.

However, experience has shown that users are easily overwhelmed by the task of spatially modeling the three-dimensional environment to make sense out of the dynamic representation. Although one should not immediately reject the validity of user-initiated data modeling within immersive environments, some contextual and technological factors might have influenced this metaphor in an unproductive way. Possibly, more user-friendly and precise interaction mechanisms and high-resolution display devices might solve many encountered problems. However, to further evaluate the potential of data-driven particles, other means of automatic data visualization need to be determined, such as local rules that regulate and stabilize the emergent dynamic events into effectively perceivable behaviors.

4.3. Galaxy

The Galaxy World application was developed in close collaboration with a private partner, who became interested in the infoticle visualization approach after a demonstration of the Modeling World scenario in the blue-c virtual reality theater. This collaborator, Ove Arup Partnership Limited, is a privately held firm, practicing in the field of consulting engineering services, especially in the built environment. At present, it is operating 71 offices in 32 countries, employing over 6500 staff members. The next paragraphs describe the visualization application that was conceptualized, programmed and evaluated during a one month visit to the Arup Research & Development group (Arup R&D) in London.

Notably, the term *galaxy* is regularly used in the field of information visualization, for instance to denote the mass of information in the Galaxy of News project (Earl, 1994) or to describe the point clusters generated by the well-known galaxy scatterplot technique (Wise, et al., 1995). However, the infoticle Galaxy World described in the following paragraphs should not be confused with these projects.

4.3.1. Goal

Arup has created internal communities of practice which they call *skill networks*. These consist of staff members who are specialized in a specific field of expertise, such a structural engineering, fire engineering, acoustics, etc., or have a professional interest in being a member of that community. Although these experts share a common corporate identity, they are often spread out geographically throughout the international offices. To enable an effective communication of project experiences and specialized knowledge, each skill network is represented by a website that is posted on the corporate *Intranet*. A corporate Intranet is an internal network architecture based on Internet standards, which users are able to access using ordinary Internet browser software. Each of these sites continuously keeps track of projects, guidelines, learning and other related issues. All employees have free access to these document databases, with the primary goal to geographically spread the knowledge that is embedded in their content. Due to the technical fact that all internal network traffic passes a single router (see Figure 4.8), complete log files are available that expose the individual use of documents, related to IP numbers, date and time.

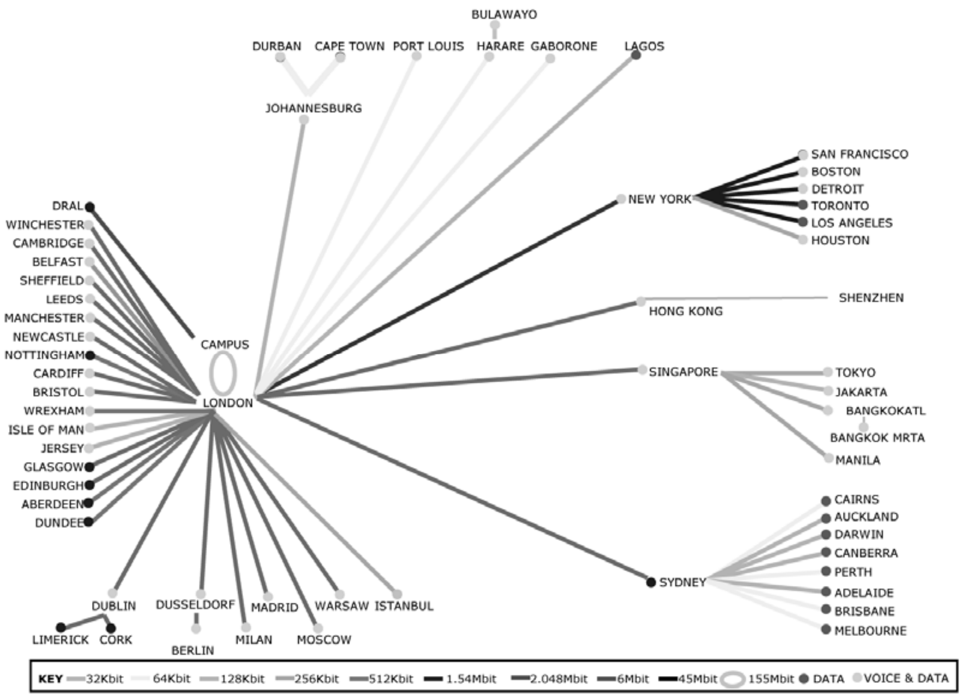


Figure 4.8. Arup Intranet network (as of October 2002).

People at Arup R&D expressed their dissatisfaction with the monthly web log reports they were purchasing from an external company. These reports primarily show quantitative rankings and standard pie charts and bar graphs, based upon the presumption that the log data used was *detailed* and principally *faultless*. Like most traditional web log analyses, rankings were based upon fixed monthly access rates, and included simple file name comparisons of the top ten documents. Furthermore, the reports presented the analyzed data by abbreviated file names and cryptic IP numbers that revealed only little informational value.

Next to some unexplainable inconsistencies Arup had discovered in the presented diagrams and rankings, they expressed their frustration about the fact that a lot of information they were interested in was still hidden. For instance, they wanted to determine whether all geographic regions were effectively accessing the internal documents, and whether the network was being effectively used as a knowledge spreading instrument throughout the whole company. They were also curious to see which specialized documents were being accessed by the correct target public possessing a relevant skill level. Additionally, they were questioning the validity of the log data and even the accuracy and correctness of the logging mechanisms.

The internal goals of Arup R&D to support the implementation of an infoticle visualization application were manifold. The company is specialized in consulting for the building industry, so that their experience in visualization is primarily focused on real-world simulations of physical datasets. However, some people inside the company are attempting to promote the potential of abstract information visualization for engineering purposes. The following goals can be distinguished.

- **Web Log Analysis.** To analyze the effectiveness of the currently used logging mechanisms and protocols. To search for hidden usage patterns in the historical Intranet web logs, in comparison with the currently acquired commercial log analysis reports.
- **Knowledge Management Tool Evaluation**. To evaluate the relatively novel method of collecting and spreading knowledge through skill-specific websites published internally on the company's Intranet network.
- **Geographic Knowledge Management Analysis.** Effectiveness analysis of the employed knowledge management network tool, by detecting the geographical usage of valuable documents throughout the international offices of the company.
- **Skills Analysis.** Discovering and analyzing the real-world usage of the implemented knowledge management tool by the specific, internationally spread skill groups.
- **Information Visualization Business Potential.** To promote internally the increasing international and economical importance of abstract information visualization. To explore the future business potential of visualizing abstract information for an engineering consulting company.
- **Virtual Reality Technology.** To experience first-hand the design, development and implementation of a real-world virtual reality application.

4.3.2. Implementation

In response to the previous context, some novel aspects of the infoticle metaphor were created, incorporating the experience gathered from the Modeling World evaluation.

– **Scenario Development.** The visualization application development was primarily targeted to the previously mentioned expectations of Arup R&D, which therefore can be imagined as the real-world *client* of the application. A new infoticle scenario, called Galaxy World, was conceptualized to visualize the Intranet network usage of the largest skill website in use, collecting relevant documents regarding *structural* engineering.

– **Data Acquisition & Filtering.** Figure 4.9 shows the standard structure of the original, raw datasets, which consisted of a large collection of ASCII text files. The acquired web logs spanned a 9 month timeframe, and contained about 850.000 hit entries. Detected data inconsistencies were removed or altered without loosing the dataset validity.

```
#Software: Microsoft Internet Information Server 4.0
#Version: 1.0
#Date: 2002-03-06 00:02:12
#time c-ip cs-username cs-method cs-uri-stem cs-uri-query sc-status sc-bytes cs(User-Agent) cs(Cookie)
cs(Referer)
00:02:12 69.132.5.6 - GET /ssn/Corus2/slimdek/slim110/design/C113.htm - 200 1388
Mozilla/4.0+(compatible;+MSIE+4.0;+Windows+NT;+Site+Server+3.0+Robot)+Ove+Arup+&+Partners+International
SITESERVER=ID=e196e01536adbfc0a19af77d5dc39ebb;+CFID=1030952;+CFTOKEN=17
00:02:12 69.132.5.6 - GET /ssn/Corus2/slimdek/slim110/design/C114.htm - 200 1387
Mozilla/4.0+(compatible;+MSIE+4.0;+Windows+NT;+Site+Server+3.0+Robot)+Ove+Arup+&+Partners+International
SITESERVER=ID=e196e01536adbfc0a19af77d5dc39ebb;+CFID=1030952;+CFTOKEN=17
00:02:12 69.132.5.6 - GET /ssn/Corus2/slimdek/slim110/design/C115.htm - 200 1389
Mozilla/4.0+(compatible;+MSIE+4.0;+Windows+NT;+Site+Server+3.0+Robot)+Ove+Arup+&+Partners+International
.......
```

Figure 4.9. Raw web log structure.

– **Data Abstraction.** All IP numbers are cross-matched with a list of known geographical IP-ranges, and then mapped onto the geographical regions of the ARUP offices. These entries are rearranged into seven geographical groups consisting of approximately equal amounts of employees. Table 4.1 shows that all documents are restructured into four main data categories, depending on the file media type and the document functionality.

Document		Structure		Image		Other	
Type	Format	Type	Format	Type	Format	Type	Format
.doc	*MS Word*	.cfm	*Cold Fusion*	.gif	*Graphic Interchange Format*	.js	*Javascript code*
.pdf	*Adobe Portable Document Format*	.asp	*Active Server Page*	.jpg .jpeg	*Joint Photographic Experts Group*	.css	*Cascading Style Sheets*
.txt .rtf	*ASCII Text*	.html .htm	*HyperText Markup Language*	.png	*Portable Network Graphics*	…	
.ppt	*MS Powerpoint Presentation*	.xml	*Extensible Markup Language*	.swf	*Macromedia Flash*		
.pps	*MS Powerpoint Slide Show*	…		.tiff	*Tagged Image File Format*		
.xls	*MS Excel*			…			
.zip	*DOS Compression*						
…							
Functionality							
knowledge, content		content, interface		interface, layout		layout, interaction	

Table 4.1. Document categories.

Table 4.2 shows that the raw data files have been transformed into a dataset that contains specific informational values, as numerical IP numbers have been transposed into regions, while documents are categorized according to their most probable end-use. In addition, the whole dataset is ordered in time, and data object reoccurrences are detected and linked.

Id	Data Identifier	Access Date	Access Time	Document Name	Data Type	Access Region
0	-	2002-07-02	12:32:59	section2.pdf	pdf/document	69.35.5.6/London
1	-	2002-07-02	12:32:59	frameset.cfm	cfm/structure	169.34.15.6/UK
2	-	2002-07-02	12:33:04	default.cfm	cfm/structure	69.35.5.6/Australia
...	...	...	...	...	...	...
16	-	2002-07-04	12:33:25	default.cfm	cfm/structure	133.2.5.126/USA
17	-	2002-07-04	12:33:25	default.cfm	cfm/structure	133.2.5.126/USA
18	-	2002-07-04	12:36:52	ref.gif	gif/image	133.2.5.126/USA
19	-	2002-07-04	12:37:19	default.cfm	cfm/structure	69.35.7.12/London
20	-	2002-07-04	12:37:23	susweb.css	css/other	133.2.5.126/USA
...	...	...	...	...	...	...
x	-	2002-0x-0x	xx:xx:xx	file name	type/category	IP/region

Table 4.2. Abstracted dataset.

– **Data Conversion.** All reoccurring data objects within the dataset are associated with unique identifiers. The whole dataset is transformed into a data typology that can be efficiently read by the infoticle data processing algorithms. Table 4.3 shows what this means in practice: all available data entries are listed in a single table, and general textual data entries become labeled with numerical values identifying different categories.

Id	Data Identifier	Access Date	Access Time	Document Name	Data Type	Access Region
0	1	2002-07-02	12:32:59	section2.pdf	1	1
1	2	2002-07-02	12:32:59	frameset.cfm	2	3
2	3	2002-07-02	12:33:04	default.cfm	2	5
...	...	...	...	...	...	...
16	3	2002-07-04	12:33:25	default.cfm	2	2
17	3	2002-07-04	12:33:25	default.cfm	2	2
18	12	2002-07-04	12:36:52	ref.gif	3	2
19	3	2002-07-04	12:37:19	default.cfm	2	1
20	44	2002-07-04	12:37:23	susweb.css	4	2
...	...	...	...	...	...	...
x	1-#names	2002-0x-0x	xx:xx:xx	file name	1-4	1-7

Table 4.3. Transformed dataset.

– **Metaphor Adaptation.** The infoticle behavior rules have been described in detail in Section 3.5.4. Infoticle – Simulation – Behavior. In contrast to the Modeling World, it was preferred to generate the visual patterns automatically and to limit the interface to contextual exploration instead of facilitating the remodeling of tools. It was observed that the acquired web log data inherently possesses specific characteristics of parallel sequential time-varying data: within a certain timeframe, multiple documents may have been accessed numerous times, and a single document may have been read never or repeatedly, and this by several regions or a single one. In short, within each database timeframe it is unpredictable which or how many documents have been accessed, by whom and for how many times. The following basic elements of the Galaxy World are illustrated in Figure 4.10.

- **Emitter.** The three recognized data type categories are represented by three separate infoticle emitters. Consequently, each of the document typologies is characterized by a different infoticle color. The emitter source is positioned in the center of the virtual representation. The source is not visually represented because the exact emitter position is no longer relevant after the initial infoticle generation, since all infoticles have an eternal lifespan.

Document Emitter	*orange*
Structural Emitter	*purple*
Image Emitter	*red*

Table 4.4. Document type color.

- **Infoticle.** Each infoticle represents a unique document that was retrieved from the web log files. Each retrieved data object within an application timeframe thus contains the document name, the document category, the access frequency within the active database timeframe and the average force that corresponds to all regions having accessed the document. Each infoticle initially swirls around the center of the scene until its corresponding document is accessed by a region and its trajectory becomes influenced by a force.

- **Force.** Each force symbolizes one of the seven different regions that contain equal employee quantities. These regions include London, United Kingdom, rest of Europe, Asia, USA, Hong Kong and Australia. Forces are represented by circular icons and are positioned around the center of the virtual representation.

- **Filter.** The Galaxy World does not contain filters in order to simplify the resulting visualization and retain representation reproducibility when manipulating the time simulation direction or speed.

- **Application Reconfiguration.** The relatively large amount of available data entries requires an intensive optimization of the data processing algorithms. As the Arup R&D group does not readily possess any virtual reality hardware for prototyping purposes, the application initially has to be adapted to a standard computer screen, computer desktop environment and traditional mouse interaction. The available hardware is reduced to a dual-processor Linux desktop machine equipped with a moderately performing graphics card. Consequently, all remote processes, such as interaction evaluation, application simulation, data querying and database processing have to be executed on a single desktop machine. In spite of these adaptations, the Galaxy World remains fully functional within virtual reality environments.

  - **Collaboration.** To facilitate future collaboration experiments, an extra circular force icon functions as virtual placeholder for the three-dimensional video representation of a remote participant.

  - **Timeframe.** The database timeframe has a length of one hour. This duration corresponds to an application timeframe of thirty frames, taking about three seconds, at which the represented data becomes updated.

  - **Interface.** A text legend at the bottom of the screen denotes the current database timeframe that is loaded by the infoticle system and keeps track of the current time direction state. Whenever an infoticle is selected, the legend shows the specific document name and a corresponding timeline trace ribbon appears in the scene. Three timeline traces can be shown simultaneously, allowing for a direct visual comparison of different infoticle trajectories and thus document usage histories.

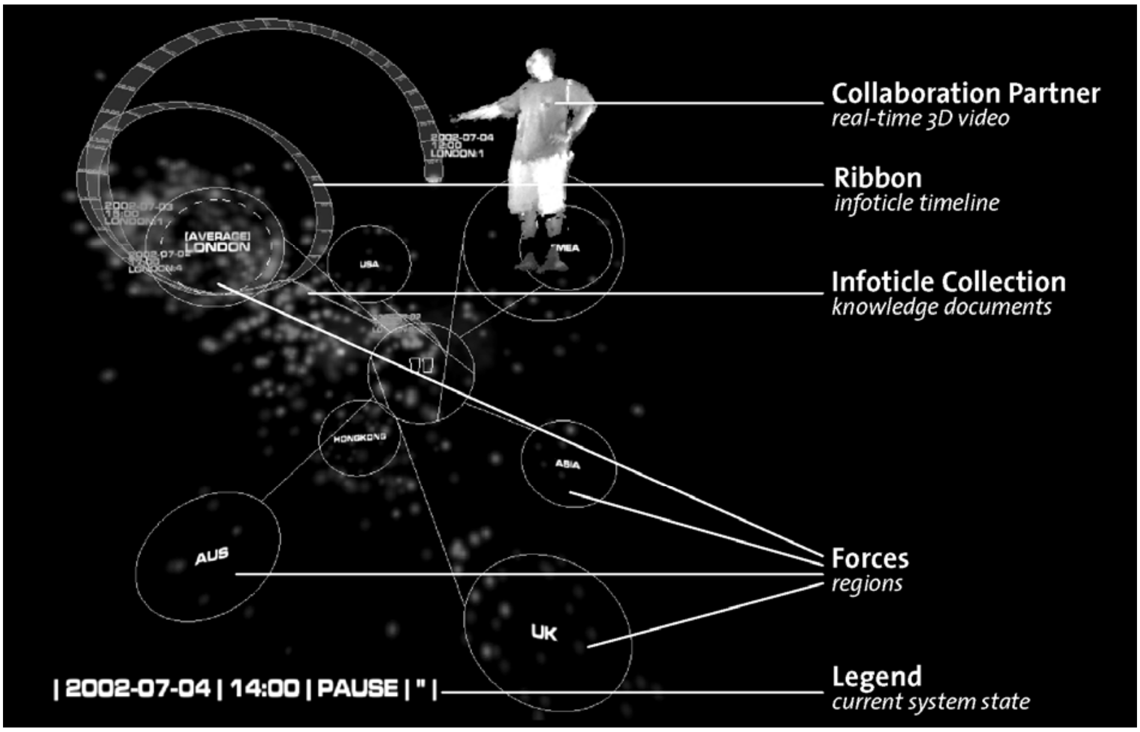


Figure 4.10. Galaxy World scene.

Figure 4.11 shows the Galaxy World project on the blue-c Powerwall and in the blue-c CAVE virtual reality installation.

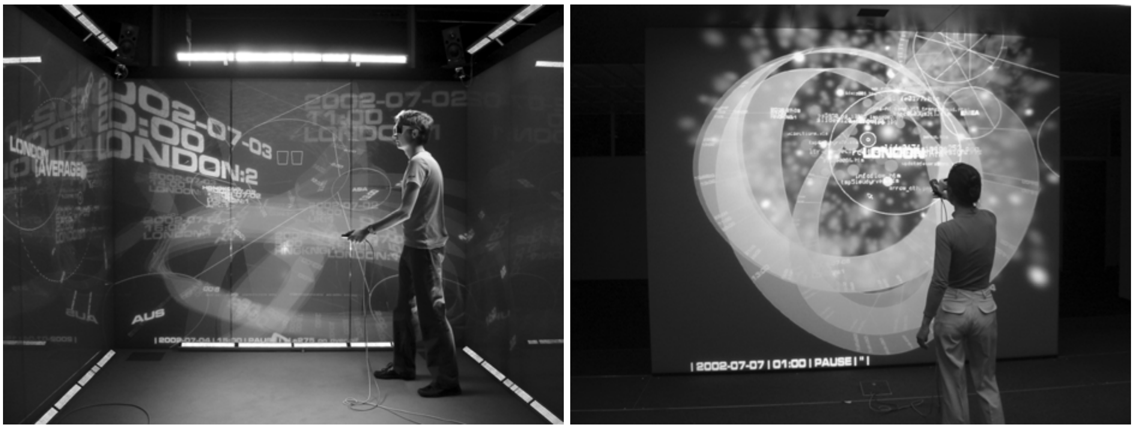


Figure 4.11. Infoticle implementation.

Later on during the development, some small-scale experiments were conducted to test potential collaboration features of the infoticle method. Figure 4.12 demonstrates the technical communication requirements between two remote participating sites. Although theoretically feasible, the computing requirements for rendering the real-time three-dimensional video inlay of the collaborating partner was found to negatively influence the framerate performance of the data-driven particles and vice versa. Furthermore, it was quickly discovered that various fundamental issues regarding the real-time simultaneous synchronization of the large amount of infoticles needed to be solved first (see also Section 3.6.4. Infoticle – Interface – Collaboration).

Figure 4.12. Collaboration experiments.

4.3.3. Pattern

In the following descriptions, a *data entry* within the web logs is considered as a single *use* of the corresponding document, although theoretically this relationship does not always hold: users might have just *clicked* on the document, or looked at it so briefly that assuming this data entry to represent an effective usage would be an overestimation. However, no additional or usable information about the document usage was available, although such data would considerably increase the representation reliability and validity.

Notably, the Galaxy scenario should not be confused with *network visualizations*, in which typically the physical transmission of unique electronic packets is represented. Instead, galaxy infoticles denote the time-varying *usage* of physically equal but mentally shared documents and the knowledge they contain, which has no physical meaning or direct visual counterpart. As a direct result of the infoticle behavior rules, different spatial behaviors can be visually distinguished. These distinct patterns can be discriminated in several modes, either by interpreting the visual features of static trace shapes or by observing dynamic infoticle behaviors. Because of obvious analogies, these phenomena are uniquely identified with terms taken from the world of physics, astrophysics and astronomy. In the next list of pattern descriptions, the Intranet files' typologies and data objects that were discovered to exhibit these particular behaviors have been mentioned as well.

- **Transfer.** Whenever the data values of an infoticle are altered, the visualization system immediately checks whether the new data values cause a change in the infoticle's average force. If so, the infoticle immediately becomes spatially directed to the proximity of that specific force, following the rules of Newtonian mechanics. The resulting infoticle trajectory resembles that of a straight line pointed towards the newly calculated average force. This sudden change in behavior can be easily recognized, as the transfer pattern is characterized by an abrupt straight trajectory.
- **Global.** The time-averaged force calculations that are based upon the document's access history produce several large, linear infoticle clusters that spatially connect pairs of forces. These groups consist of documents that are shared extensively by two or more regions. Changes in size, density and behavior of these infoticle clusters can be easily detected and traced in time.
  - **London - UK.** Figure 4.13 shows that the most recognizable global pattern connects the London and UK and also, but less, the London and Hong Kong regions. London is, not surprisingly, the city where the corporate headquarters and campus are located.

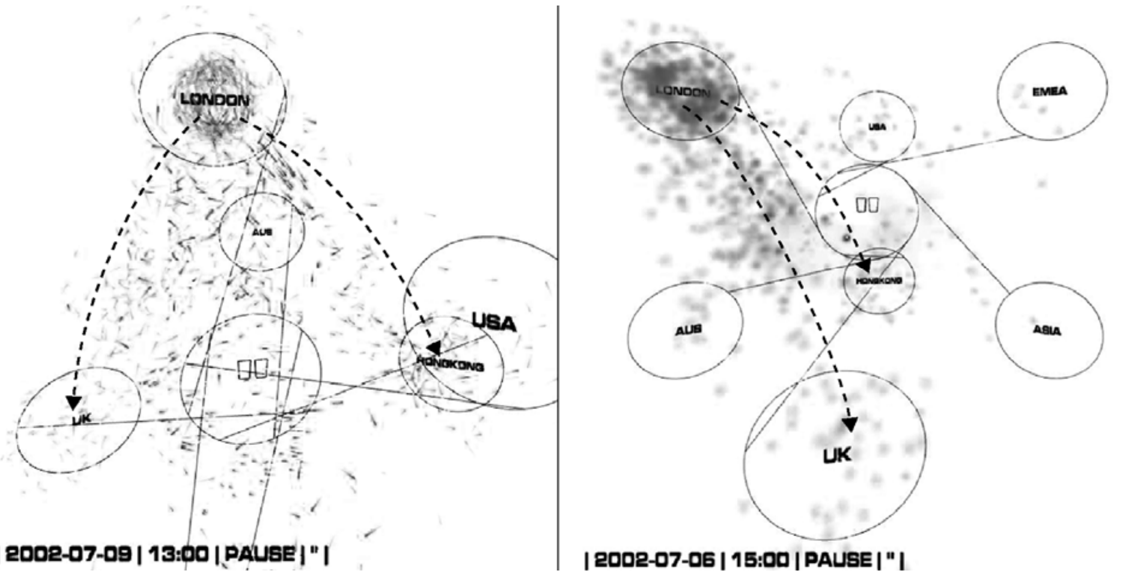


Figure 4.13. Global pattern.

- **Time-Zone.** Due to the international nature of the Intranet network, a time zone shifting influence can be perceived.
  - **Time of Day.** Document usage peaks at office opening and closing hours and around lunch time. A previous, independent investigation had concluded that many employees surf to websites that are not related to work. A comparison of Intranet with Internet or e-mail usage could further clarify these findings, and demonstrate the proportion of work-related web surfing.
  - **Day of Week.** Usage alterations during the weekend or at the end of the week are easily detected by observing the relative quantities of infoticle flocks lighting up and changing direction exactly at those points in time (Figure 4.14).

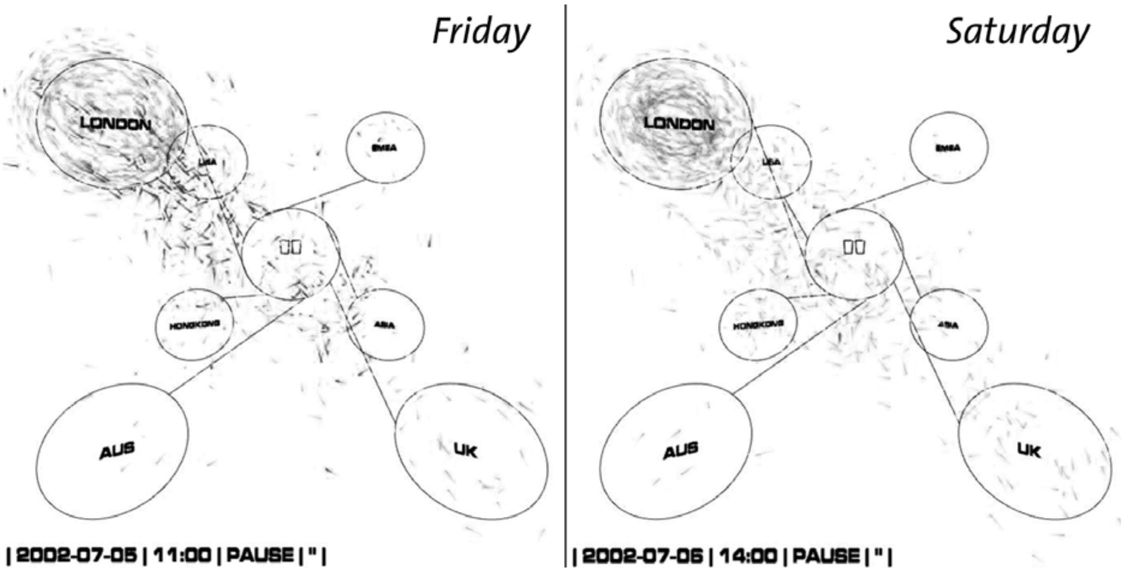


Figure 4.14. Time-Zone pattern.

Although these time-related patterns prove that the spatial behavior rules are able to visually represent the characteristics of time-varying data, some data patterns might be missed by the continuous time zone shifting within the dataset. For instance, some usage trends within shorter timeframes that are internationally similar could possibly become spread over a one day timeframe. A potential solution consists of converting all timestamps of the acquired data entries to a single, common time zone.

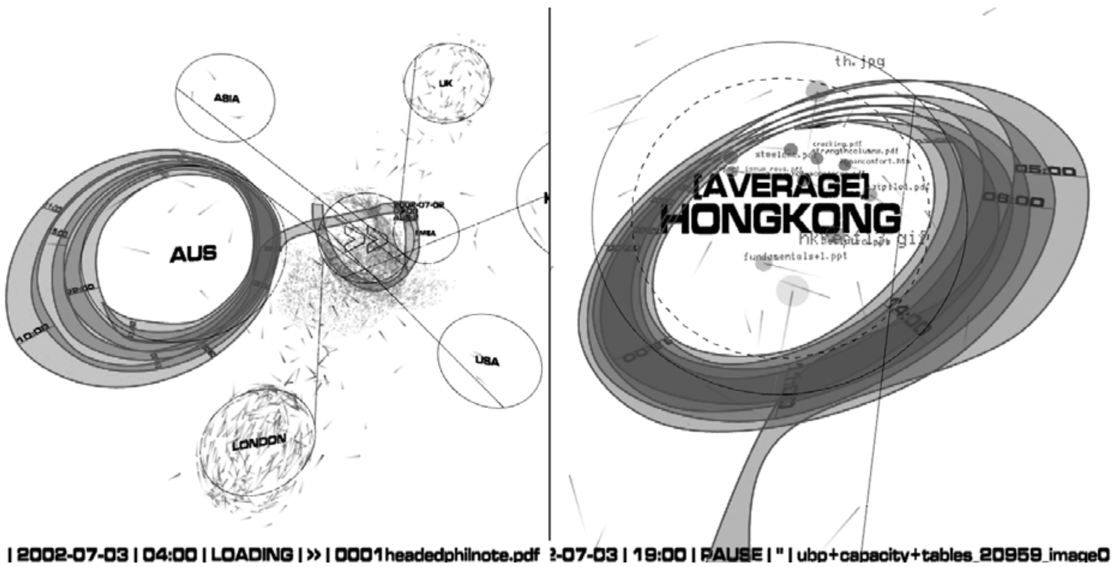


Figure 4.15. Star pattern.

– **Star.** An infoticle spinning around a force in a circular, periodic orbit, as shown in Figure 4.15. A *star infoticle* represents a document that has recently been accessed, just reached the average force and started to orbit around it. During this process, it is slowed down by the continuous drag influence and moves relatively far from the average force. A star infoticle is highly sensitive to any further direction or speed alteration by the fragile nature of its trajectory, as these will break the perfectly circular curve. Notably, the star pattern might be the only recognizable visual pattern generated by the system for many documents. In practice, this means that these documents have been accessed once, probably by their own creator, and are not consulted by any user or region afterwards.

  – **All.** One cannot denote a single type of documents that exhibit a star behavior, as this visual pattern is a time-delimited process all other patterns have to pass through first.

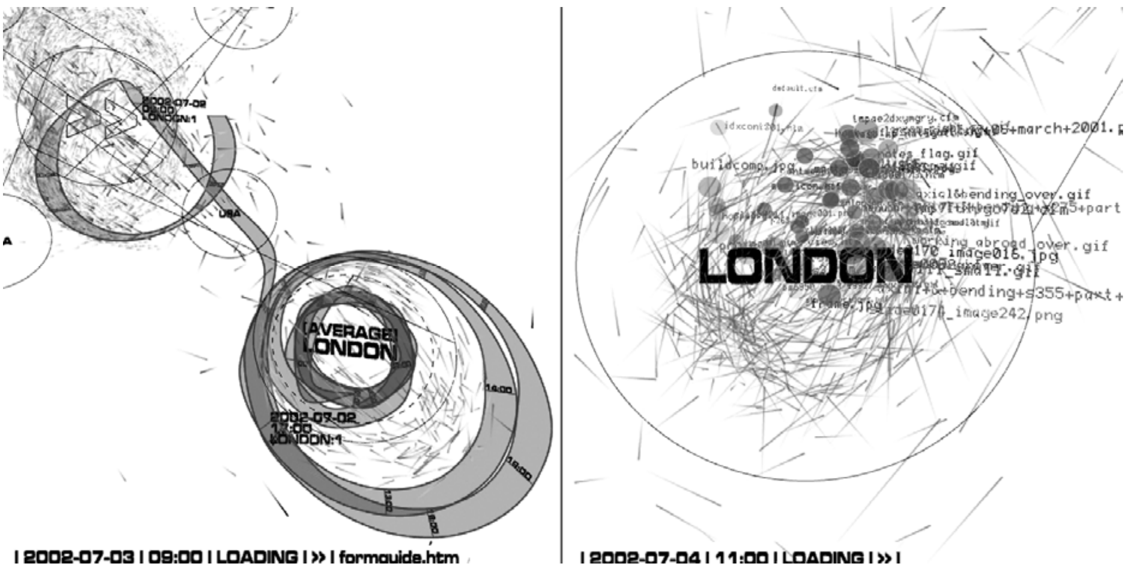


Figure 4.16. Electron pattern.

– **Electron.** An infoticle falling back to a force due to the continuous speed decrease caused by the drag influence. Figure 4.16 shows that an *electron infoticle* has a circular trajectory, and is close to the force. A typical electron infoticle represents a document that was accessed a reasonably long time ago, and thus might contain *non-shareable, outdated* or *redundant* knowledge.

  – **Redundant Knowledge.** An electron represents a document that has been accessed at least once a reasonably long time ago compared with the active database timeframe. In practice, this pattern illustrates that the document is not shared within its own or other regions. Such a document might thus contain redundant knowledge that does not fit the purpose of the internal knowledge network, although it occupies a certain amount of computing resources. However, the infoticle system does not quantify the duration of the inactive timeframe nor forms an objective measurement about any document's content. In practice, it is up to the network administrator to determine the importance of electron infoticle documents before removing them from the knowledge management Intranet system.

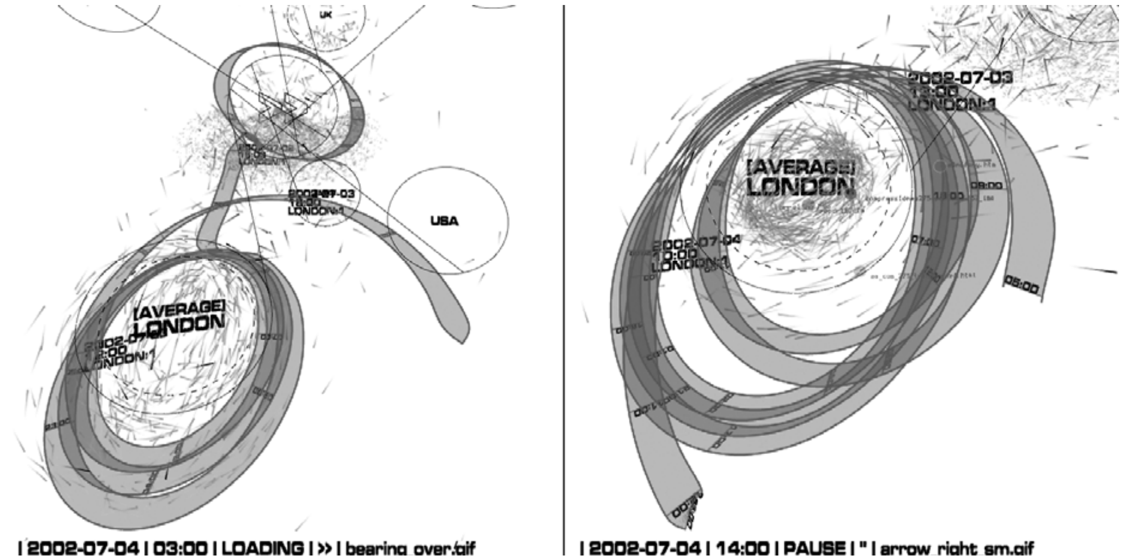


Figure 4.17. Comet pattern.

- **Comet.** An infoticle that is accessed more than once by a single region, or same set of regions, within a short period of time. Due to the considerable increase in speed and the absence of significant force direction changes, the resulting trajectory becomes elliptical and quasi-periodic, as demonstrated in Figure 4.17. Such an infoticle typically visualizes a document that is being shared by the same region or the same set of regions.
  - **Single User Access Burst.** Some infoticles leave the virtual scene, or are moving at the outskirts of the representation. These data-driven particles represent documents that were accessed frequently within a very short timeframe by a single region, resulting in a relatively high increase in speed and no change in direction. Further research has proven that such behavior is often caused by mistakes in the utilized web logging mechanism. For instance, one such infoticle represents an Adobe PDF document that was accessed 72 times by a single IP number within 3 seconds. The most plausible explanation for this behavior is a single person who is browsing through the 72 pages of reasonably (+10MB) large, and thus slowly loading document in a very rapid manner, while Adobe Acrobat Reader attempts to cache the ever-changing active page for each successive page number alteration. Note that this particular document receives a fairly high, but fully inaccurate, ranking in a traditional *most popular* web analysis tool because of the relative high number of file accesses.

Figure 4.18. Burst pattern.

- **Burst.** A cluster of multiple infoticles that change their direction simultaneously and travel in the same direction, similar to Figure 4.18. Such infoticle *bursts* can be easily detected, as they take up a considerable part of the representation space and contain a large number of individual data-driven particles with equal direction. An infoticle burst generally consists of documents that are conceptually related in terms of *content*, such as PowerPoint presentations saved for the web, or groups of Adobe Acrobat PDF documents of the same project or subject.
  - **PowerPoint Presentation.** One can observe the sudden emergence of bunches of infoticles representing documents, images and style sheets that are directed towards a single force. Most often, these clusters represent a series of files from a PowerPoint presentation that has been saved for the web. Storing such presentations for the web typically generates dozens of unique web pages, labeled with a sequence number (e.g. slide00101.htm). Consequently, browsing through such a presentation requires the loading of many successive web pages and individual images. Notably, individual PowerPoint presentations are unlikely to show up in a traditional popularity web log ranking, as they have been divided in many separate, atomic parts. The infoticle data representation method, on the contrary, is able to visualize a single visit to a PowerPoint presentation as a whole.
  - **Project Subject.** At some points in time, several PDF documents become animated in a burst pattern. Often, this behavior is caused by a single user who is interested in a specific specialized subject, and browses through the related documents in the knowledge network. Consequently, *burst infoticles* exhibit some relationship in document content, although often the document names do not readily reveal any particular conceptual similarity.

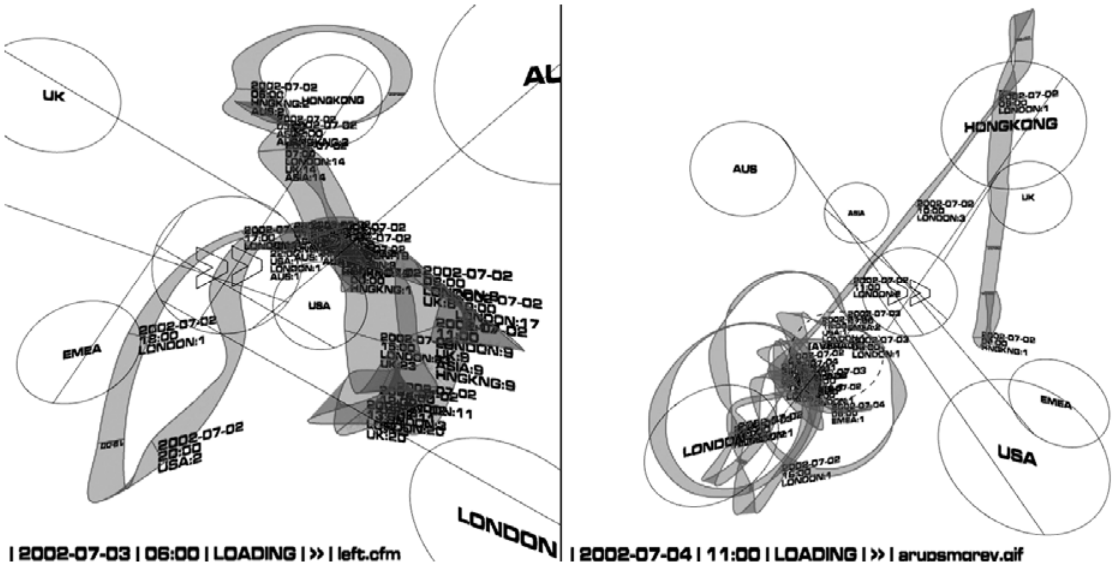


Figure 4.19. Quark pattern.

– **Quark.** An infoticle that behaves erratically, changing directions multiple times within a short time span, as shown in Figure 4.19. This chaotic behavior is easy to spot, as this motion characteristic visually pops out. *Quark infoticles* represent files that are effectively shared across geographical regions. Typical data-driven quarks are portal documents, or individual documents that became popular by internal advertisements. Because quark infoticles are influenced by rapidly changing regions and thus attracted by constantly altering forces, their irregular trajectory is situated in a spatial zone in between the regions and contains many fluctuating points of change.

  – **Portal Document.** Portal documents are web pages that offer access to more detailed and lower-hierarchical information and are typically named 'index.html' or 'frame.cfm'. Usually, these documents are accessed multiple times by different regions over large periods of time. Portal documents are relatively easily recognizable by their chaotic spatial behavior and corresponding timeline ribbon.

  – **Internal Advertisement.** Some documents are advertised on portal documents in the context of specific internal events or important project fulfillments. These documents are accessed many times, although this phenomenon is relatively limited in time.

In general, Galaxy infoticles change their pattern behavior in a stable way, even when they are exposed to a rapid sequence of data updates. Accordingly, Figure 4.20 shows how the continuous change in average force position results in a steady succession of interpretable infoticle behavior patterns.

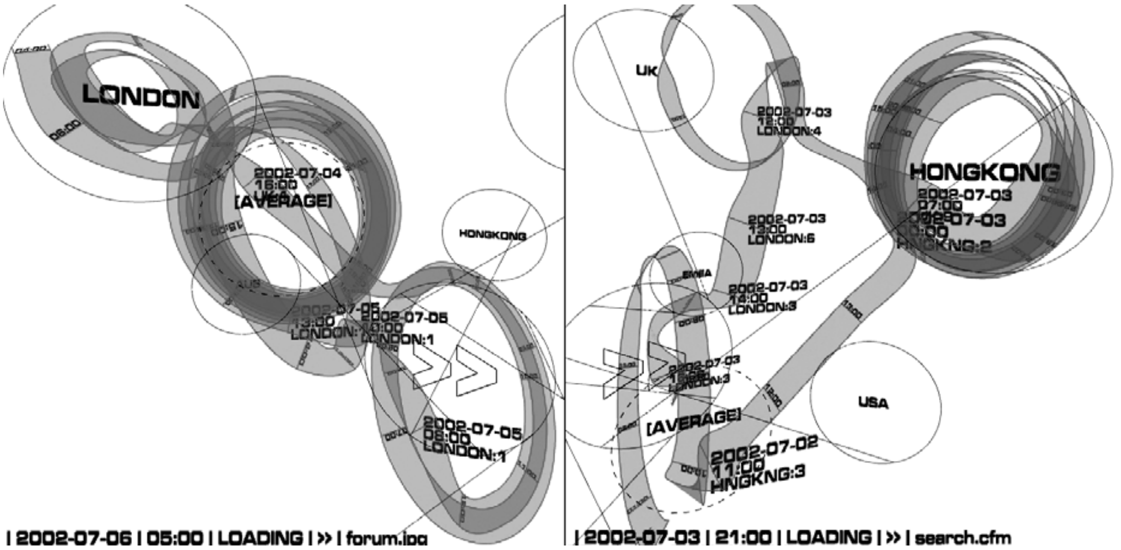


Figure 4.20. Pattern evolution.

Notably, the previously mentioned patterns can be detected in different contexts, depending on the spatial proximity to the scene center or the actual time simulation state. The exact pattern detection methodology and the visual semantics behind the Galaxy World pattern generation are analyzed in Chapter 5.

4.3.4. Evaluation

The visualization application client, in this case the people at Arup R&D involved in this project and the audience attending the various presentations at Arup, was clearly satisfied with the visualization prototype that was presented.

The client particularly liked the *non-conclusive* approach, leaving judgments about document content and website performance over to the visualization end-users instead of presenting them with a static collection of possibly faulty, automatically produced rankings that are irrelevant in the context of data exploration. Furthermore, they started to be interested in *odd-performing* documents, a phenomenon they were not aware of before. People who were already familiar with the dataset, such as the knowledge network administrators, particularly appreciated the *interactive* aspect, allowing them to create their own data queries by placing the regions into different spatial settings or to focus on specific timeframes. For the first time, they could finally influence the exploration process instead of receiving a static and uncontrollable analysis.

Many remarks were given about the *added value* of this tool for an engineering consultancy company as Arup, where web log data or even abstract data in general currently does not play an important role. Accordingly, it was observed that the intrinsic value of web log datasets, and therefore also the value of the whole visualization application, did not appeal to all observers. However, such datasets could potentially be used for judging the investment performance of the recently employed knowledge management tool. Some adoption barriers could be detected as well: it was discovered that the people who were enthusiastic about the application mainly had professional or educational backgrounds situated in abstract disciplines, such as architecture, research, finance and computer science. In contrast, people such as engineers, who typically deal solely with a physical reality context on a daily basis, seemed to have considerable difficulties to fully comprehend the dynamic nature of abstract data. Consequently, many analogies were made with traffic or smoke distribution simulations. However, such visualizations are based upon fundamentally different principles, such as mathematical influence matrices and thermodynamic dispersion laws.

Regarding the visualization itself, the initial test users - mainly consisting of colleagues and people who had already some experience with web log datasets - were able to easily perceive the different visual patterns and linked them metaphorically to the various sorts of documents. The combination of time-varying data updates with the pre-attentive feature of motion enabled users to evaluate unexpected data events, such as suddenly popular documents or geographic usage connotations. Simultaneously, long-lasting trends, such as the performance of individual documents, could be observed as well.

The infoticle visualization was able to uncover data patterns that were previously unknown. The document type color coding unmasked relatively more structural files as quarks, proving that the website itself was accessed by most regions, in contrast to the document files, and so the knowledge. Many documents enjoyed a limited popularity in short timeframes, a phenomenon probably caused by actual projects or hot research topics at those points in time. Adobe Acrobat Portable Document Files seem to generate errors within the log mechanism, making the corresponding knowledge documents automatically and wrongfully more popular in traditional log analyses. A huge discrepancy was detected between the London versus United Kingdom and London versus Hong Kong regions in comparison with the other offices in the world. The company especially appreciated the fact that the infoticle method could not put an exact quantifiable number on such unbalanced phenomena, so that the corresponding information representation could still be presented to responsible persons in an unbiased way.

The application seemed to produce a high degree of user engagement, as people liked to use it for longer periods of time and expressed their appreciation for the overall interface design. Users especially appreciated the direct and simple user interaction, which stimulated the exploration of the dataset at different levels of detail. In contrast, some users, mostly inexperienced in the world of computing, seemed to be visually overwhelmed, possibly by the combination of the immersive virtual reality characteristics and the continuously dynamic visualization features. It was also noticed that many persons without any virtual world experience expressed problems with effective navigation and interaction. The trackball navigation was cumbersome on normal desktop displays, although this was often solved after a relatively short period of training.

Users also rapidly started to invent alternative visualization scenarios that could be performed with the data-driven particle representation. For instance, they expressed their wish to be able to track the *sharing performance* of their own created documents in time or, in contrast, to trace the documents that were extensively used by their region to be sure they did not miss any important geographically-related information. These ideas can be easily accomplished by marking such documents in a contrasting color, or zooming in on spatial areas of the information representation. Notably, this visualization has the intrinsic ability to merge fundamentally different datasets within the same scene: for instance, regional document usage can be combined with internal phone calls or e-mail traffic, employee skills, job titles or even project names. However, the attempts to acquire some of these more valuable datasets were rejected because of security and privacy considerations.

4.3.5. Conclusion

The infoticle system is able to effectively visualize an Intranet log file of one month, containing a total of 128.000 Intranet log entries, generated by about 4.654 unique files. The visualization simulation updates the data values in a time-ordered way, as it retrieves and visualizes new data objects within a timeframe of one hour with a tempo of about three seconds. Data-driven particles adapt to the continuous change of average forces in a stable and robust manner. Moreover, many infoticles do not exhibit a single behavior, but instead are able to switch and adapt their visual patterns at a fast pace.

The infoticle metaphor proves to be a viable visualization approach that is capable to overcome the typically inaccurate nature of web usage data analysis tools. The system was able to process a relatively large, parallel sequential time-varying dataset, while enabling users to interactively explore the dataset during a timeline simulation. This technique also facilitated the detection of several data patterns that were not discovered before with traditional analysis methods.

Unlike most other web log visualizations, the infoticle method also represents those documents that have not or rarely been used, or are accessed collectively and simultaneously, as these time-dependent phenomena have a meaningful value in the knowledge management network context. Consequently, the infoticle representation urges knowledge network administrators to consider and deal with underperforming files or unexpected usage aspects of the management system, instead of solely focusing on the best performing documents.

This prototype proves that data-driven particles are especially suited for early information exploration purposes, when little or nothing is known about the dataset. Various data patterns that have a meaningful value emerge on both a global and an individual scale, and can be easily interpreted in an enjoyable way.

4.4. Electron

The Electron World is able to represent multi-dimensional datasets by visualizing the time-varying flow of one data dimension to another. Users can alter the direction and the speed of the timeline simulation, and may switch the active data dimension in real time. By tracking the directional changes and dynamic infoticle behaviors, the dependencies between the different data dimensions become visualized in a direct and intuitive way. This visualization scenario demonstrates one of many possible application purposes by focusing on effective *information presentation* instead of information exploration.

4.4.1. Goal

Flexible and versatile information visualization metaphors are capable to visually convey certain data tendencies to a non-expert audience. Such visualization methods must be able to effectively simulate and communicate dataset trends, and differ from expert systems that solely focus on facilitating a detailed exploration process.

Many time-varying datasets contain several simultaneous data dimensions that users want to effectively understand. The visualization purpose is then to present the dynamic evolution of these dimensions, or to show the dependencies between them. Visualizing multiple data dimensions simultaneously is not trivial. In theory, the attraction strength of each force could be correlated to a specific mathematical combination of several data dimensionalities, so that the resulting infoticle behaviors are determined by local interactions of multiple data attributes at once. However, research in cognitive science has proven that users are rarely capable to relate movements to more than one simultaneous consideration, as the comprehension complexity increases dramatically with each additional influence inclusion. Therefore, it has been chosen to determine the infoticle movements by a single data dimensionality only. Subsequently, data dimensions can be *switched off* and *on* by activating different sets of forces representing a single data dimension. As only one data dimension can be active at a time, non-related data attributes do not intervene in the resulting infoticle behaviors.

Financial datasets typically contain several data dimensions that represent some kind of time-dependent flow. Financial budgets, for instance, contain information about the original money acquisition, purpose and final expenditure. In a real-world situation, these stages generally occur successively in time, and can thus be represented by data-driven particles.

4.4.2. Implementation

The Electron scenario represents the data accumulated in several ETH-Zurich financial year reports. Several different implementation issues had to be accomplished in order to adapt to the specific dataset characteristics.

- **Scenario Development.** The dataset comprises the yearly financial budgets of the Swiss Federal Institute of Technology from 2001 until 2003, which are publicly available in the year reports published by the university. This dataset has to be effectively presented to a non-expert audience to demonstrate the evolution of the budget division.
- **Data Acquisition & Filtering.** The yearly datasets are retrieved electronically, and contain the amount, source and destination of the yearly budgets, as shown in Figure 4.21. In fact, the infoticle method is perfectly capable to visualize the precise flows in time, from source to destination, of the respective budgets. Unfortunately, the exact trajectory of each individual part of the acquired budget is unknown as all money is collected into a single, common fund. As a result, the money streams need to be mathematically simulated by switching force sets instead of exactly tracking and recreating real-world data. As all data contained numerical data values, data filtering was unnecessary.

AUFTEILUNG DER AUSGABEN NACH FACHBEREICHEN

(Angaben in 1000 CHF)		Mittelherkunft / Kreditquellen (Finanzierung)			Verwendung der Mittel nach Ausgabearten		
	Rechnung 2001 Total	1. Budget-mittel Staats-rechnung	2. Dritt-mittel	3. Mittel anderer Bundesstellen	Personal [1)]	Sach	Investitionen
BAUWESEN UND GEOMATIK							
Total	**88.563**	**78.214**	**10.349**	**0**	**76.778**	**9.601**	**2.185**
INGENIEURWISSENSCHAFTEN							
Total	**178.782**	**141.676**	**37.106**	**0**	**141.554**	**24.867**	**12.361**
NATURWISSENSCHAFTEN UND MATHEMATIK							
Total	**204.282**	**171.196**	**33.086**	**0**	**152.524**	**34.052**	**17.705**
SYSTEMORIENTIERTE NATURWISSENSCHAFTEN							
Total	**137.629**	**113.377**	**24.252**	**0**	**114.603**	**16.665**	**6.361**
ÜBRIGE WISSENSCHAFTLICHE UND INTERDISZIPLINÄRE AUSGABEN							
Total	**41.903**	**29.842**	**12.061**	**0**	**31.437**	**8.110**	**2.357**
Total Departemente und Zentrale wissenschaftliche Dienste	**753.528**	**626.291**	**127.236**	**0**	**563.471**	**141.660**	**48.397**
Total Verwaltung, Infrastruktur, Allgemeine Ausgaben	**185.430**	**178.405**	**3.632**	**3.393**	**109.991**	**72.703**	**2.736**
Total Betriebsausgaben	**938.958**	**804.696**	**130.869**	**3.393**	**673.462**	**214.362**	**51.133**
Gesamttotal Ausgaben	**1.070.260**	**935.997**	**130.869**	**3.393**	**673.462**	**214.362**	**182.435**

Figure 4.21. Financial dataset (2001).

- **Data Abstraction.** The most complex data abstraction issue relates to determining the most probable flow from the acquisition to the expenditure stage for each unit of the budget. As this information is not readily accessible, one has to assume that all acquired money is first collected into a single, shared fund and subsequently becomes fairly redistributed. Consequently, the separate money flows need to be mathematically reconstructed by simple statistical calculations. For a possible *acquisition source* A_i (e.g. financial contributions from the federal government, third-party funding, funds from other federal institutions) and *expenditure purpose* X_j (e.g. staff, material, investments) the budgetary flow is calculated as follows.

$$X_{i,j} = X_i \cdot \frac{A_j}{\sum_j A_j}$$

In practice, this means that each expenditure purpose X_j is divided in proportion to the corresponding acquisition sources. Table 4.5 shows the data structure after this abstraction process, with amounts expressed in 1.000.000 CHF. To simplify the resulting visualization, the necessary amount of forces is limited to seven different categories, by merging the groups of related departments.

TABLE OF MONEY FLOW FROM ACQUISITION TO EXPENDITURE (2001)

ACQUISITION	FEDERAL GOVERNMENT			THIRD PARTY			OTHER FEDERAL INSTITUTIONS		
EXPENDITURE Department Category	Staff	Material	Investments	Staff	Material	Investment	Staff	Material	Investments
CONSTRUCTION & GEOMATICS	67.8	8.5	1.9	9.0	1.1	0.3	0.0	0.0	0.0
ENGINEERING SCIENCES	112.2	19.7	9.8	29.4	5.2	2.6	0.0	0.0	0.0
NATURAL SCIENCES & MATHEMATICS.	127.8	28.5	14.8	24.7	5.5	2.9	0.0	0.0	0.0
SYSTEM-ORIENTED SCIENCES	94.4	13.7	5.2	20.2	2.9	1.1	0.0	0.0	0.0
OTHER SCIENCES & SPORT	22.4	5.8	1.7	9.0	2.3	0.7	0.0	0.0	0.0
CENTRAL SCIENTIFIC SERVICES	41.9	43.5	6.7	4.7	4.9	0.8	0.0	0.0	0.0
ADMINISTRATION & INFRASTRUCTURE	105.8	69.9	2.6	2.2	1.4	0.1	2.0	1.3	0.1

Table 4.5. Abstracted financial dataset (amounts in 1.000.000 CHF).

– **Data Conversion.** Each unit of 1.000.000 CHF corresponds to a single data object and thus to a corresponding infoticle. To simulate an accurate yearly budget evolution, the system needs to check for each successive year whether the 'same' 1.000.000 CHF has already flowed from acquisition to expenditure. In practice, all equal data-driven behaviors out of different timeframes are mapped onto an equal data object identifier, so that during a data update, these infoticles will stay unchanged and *constant*. In fact, in this dataset, the majority of data objects reoccur equally during most database timeframes, as the financial distribution changes slowly and minimally for each year. Table 4.5 thus shows that for a database timeframe of 2001, 106 different infoticles move between the federal government acquisition, the departments of administration and infrastructure and staff expenditure. Table 4.6 gives an impression of the resulting data table structure that is readily interpretable by the infoticle system.

Id	Data Identifier	Date	Department	Acquisition	Expenditure
0	1	2000	CONSTRUCTION/1	government/1	staff/1
1	2	2000	CONSTRUCTION /1	government/1	staff/1
2	3	2000	NATURAL SC./3	third party/2	material/2
...	...	...		...	...
16	2	2001	CONSTRUCTION /1	government/1	staff/1
17	3	2001	NATURAL SC./3	government/1	material/2
18	12	2001	NATURAL SC./3	other/3	investments/3
19	16	2001	NATURAL SC./3	third party/2	staff/1
20	44	2001	NATURAL SC./4	government/1	staff/1
...	...	...		...	...
x	x	2000-2003	1-7	1-3	1-3

Table 4.6. Converted financial dataset.

- **Metaphor Adaptation.** The infoticle metaphor needs to be adapted so that data dimension changes within the same database timeframe and the evolution of database timeframes within the same data dimension both become easily perceivable.
  - **Emitter.** The three different budget source acquisition types are depicted by three different, contrasting colors. Like in the Galaxy World, the exact spatial positions of the emitters are irrelevant for the visualization outcome as no infoticles are removed from the scene during the simulation.
  - **Infoticle.** Each infoticle represents 1.000.000 CHF, and corresponds to a data object with the following data attributes: timestamp, budget source, budget expenditure and departmental category. Infoticles are visualized by long, gradually colored *splines* instead of small, straight lines, to better represent the global directionality and long-term evolution in time.
  - **Force.** Figure 4.22 shows the three different force collections, each representing a separate *dataset dimension*. The first set denotes the three acquisition sources, the second set the seven departmental categories, and the last set the three expenditure purposes.

Figure 4.22. Electron world force sets.

- **Application Reconfiguration.** The infoticle metaphor has been significantly changed in the ability to switch data dimensions in an active state. In practice, specific force sets are turned on and off, so that the dynamic infoticle flows, and thus money transfers, can be traced in both space and time. This application scenario does not suffer from serious computational performance problems as the amount of required data updates is relatively limited. Tools such as trace ribbons do not play an important role in clarifying the representation, as it is more intuitive to iteratively manipulate the time simulation and observe the unique infoticle changes than to compare shape formalities.

4.4.3. Result

The resulting data representation consists of 1010 infoticles, collectively corresponding to a budget flux of 1.010.000.000 CHF, which initially swirl in the center of the scene. In fact, most infoticles are not subjected to any significant time-varying evolution and thus show no noticeable behavior when animating the timeline. In practice, such infoticles signify the large unchanged proportion of the yearly budget distribution: instead, the relatively small changing money flows from and towards the departments become easily perceivable. All infoticles are divided by three separate colors, each denoting a unique acquisition source. The system offers two main user interactions: altering the data dimension or the time direction.

– **Data Dimension Alteration.** After a data dimension alteration, all infoticles will immediately adapt to a new force set. Accordingly, Figure 4.23 shows how the infoticle collection becomes spatially divided when the department data attribute is initialized. Although such data dimension switching is not suited to express the exact quantities of infoticles in each cluster, the visual differences in overall *cluster volumes* can be easily interpreted. For instance, Figure 4.23 clearly illustrates the relatively small budget sizes of 'Other Sciences & Sport', 'Central Scientific Services' and 'Construction & Geomatics' in comparison with the other departmental categories. Simultaneously, users can observe the color proportions within separate clusters. For instance, the color images (see Appendix C) show that 'Administration & Infrastructure' is funded with federal government money only, whereas 'System-Oriented Sciences' and 'Other Sciences & Sport' gather a large proportion of their budget via third means.

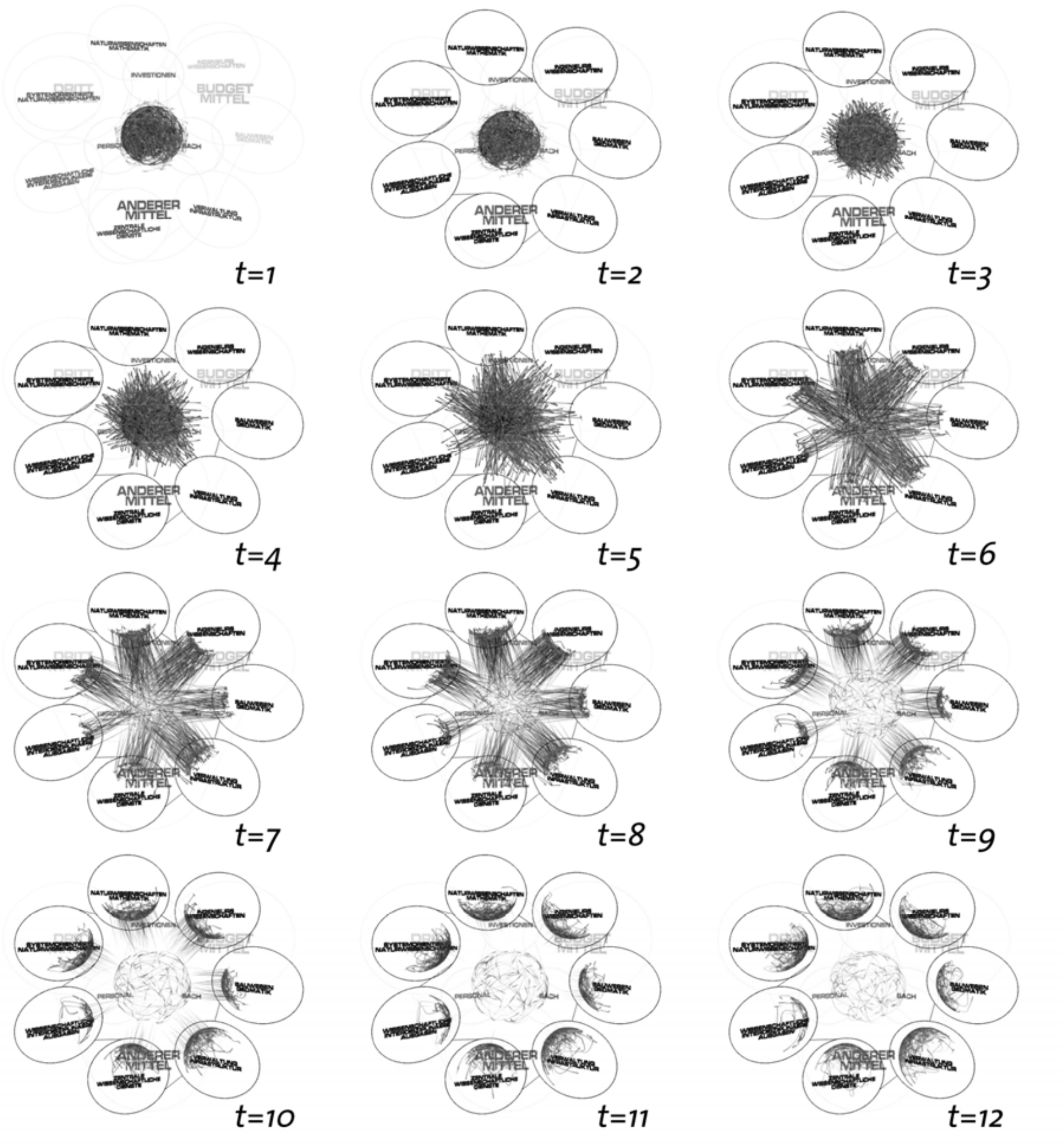


Figure 4.23. Data dimension initialization.

Figure 4.24 shows how a change in data dimensionality can have the meaning of a time-varying flow when the infoticles start from another data dimension. In this case, the data dimension is changed from expenditure to the department data dimension.

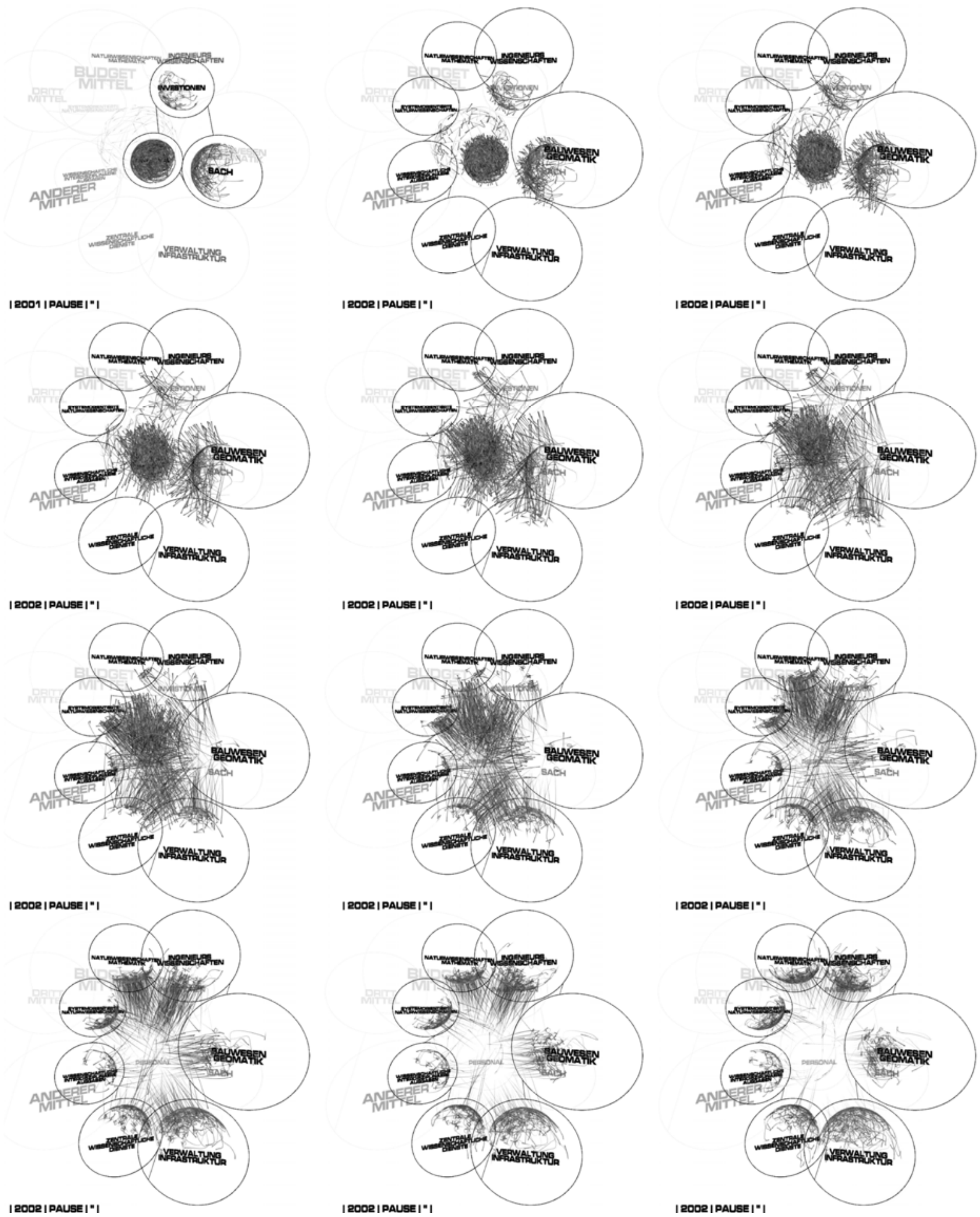


Figure 4.24. Data dimension alteration.

– **Timeline Direction Alteration.** One of the most typical dynamic characteristic of financial budgets is that they show little change over time. In fact, most budget adaptations are hidden in small numerical alterations, or typically require many different static bar chart diagrams to represent the various data dimension evolutions. Electron infoticles, however, are capable to bundle time as well data dimension changes while facilitating a continuous overview over the whole dataset. Figure 4.25 shows how the data-driven particles of the department dimension mode are animated from year 2001 to 2002. Such a change typically requires an animation of about 8 seconds, a timeframe during which the representation becomes dynamic in nature and is the most meaningful for user evaluation and interpretation.

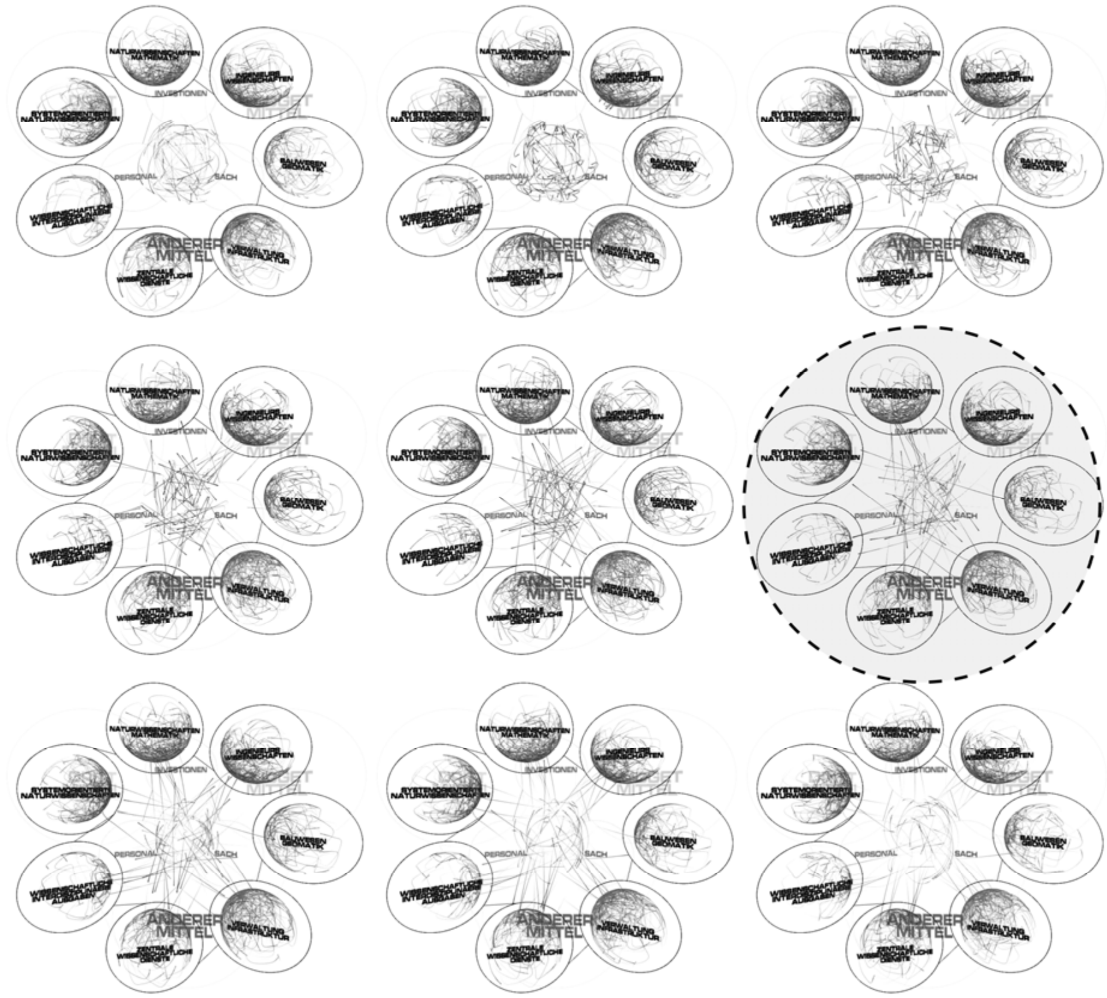


Figure 4.25. Timeline alteration (2001 - 2002).

To convey the informational meaning of the emergent data patterns in more detail, Figure 4.26 displays a close-up image from the previous timeline simulation, namely the encircled representation of Figure 4.25. This figure contains highlighted and grouped trajectories that denote the purple-colored infoticles in between the forces, hereby representing those specific parts of the budget that have been acquired from third parties and have quantitatively changed from 2001 to 2002. Please note that these clarifying graphical measures are not required when observing the corresponding color image (see Appendix C).

Due to the larger force strength in the center of the scene, the infoticles that moved *away* from the forces reached a slightly higher speed and thus already ‘arrived’ at the center cluster in this specific figure. This speed-difference has been implemented for reasons of clarity, so that two separated phases, namely the *incoming* (added to) and *outgoing* (subtracted from) budgets from the point of view of the departmental categories, can be perceived in sequence. In short, Figure 4.26 thus only shows the *additional* budgets from third parties for each of the departments in the year 2002.

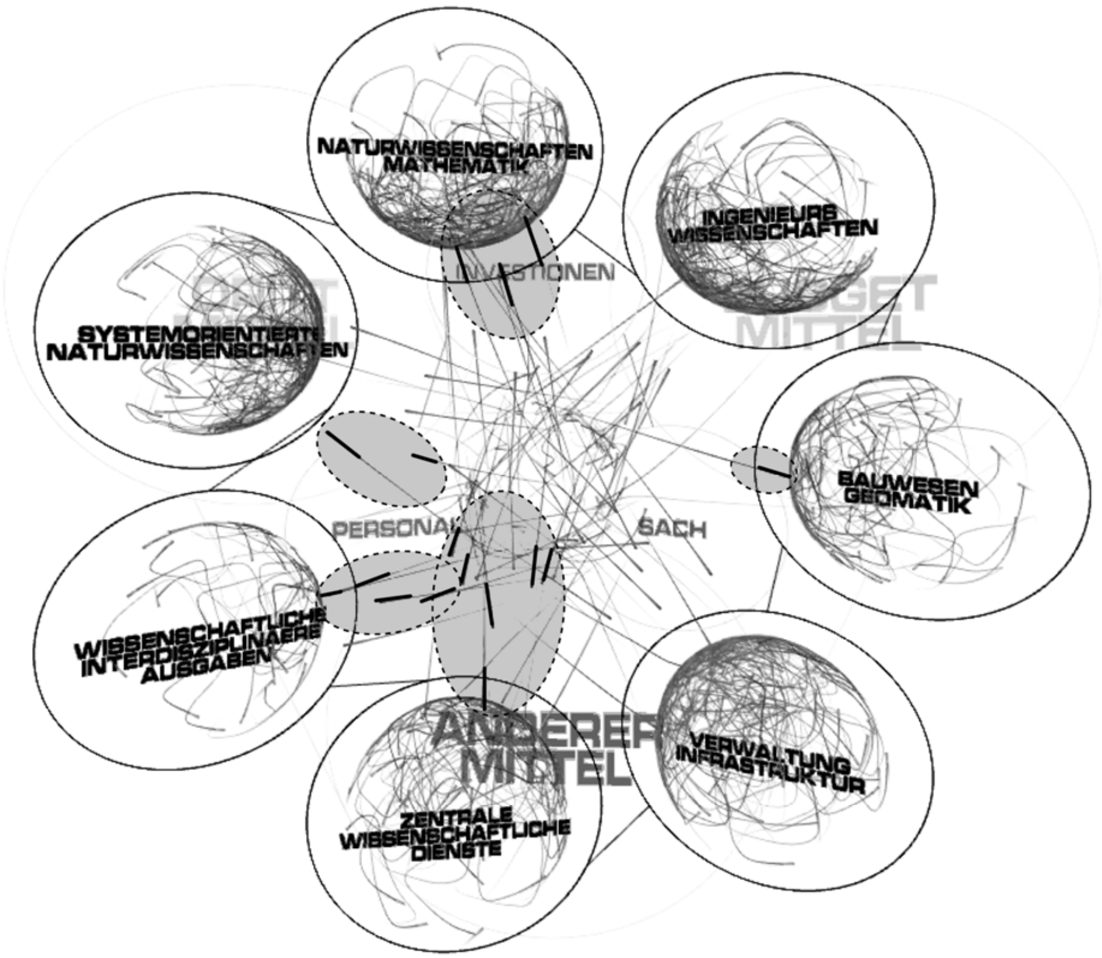


Figure 4.26. Third party evolution (2001 - 2002).

Table 4.7 demonstrates the informational meaning and numerical validity of the patterns emerging in Figure 4.26. It shows a small subset of the whole available budget, namely those parts acquired by third parties for each departmental category for the year 2001 and 2002. The number of infoticles attracted by each departmental category corresponds exactly to the gains, expressed in 1.000.000 CHF, which each departmental category has received. Because of mathematical rounding inconsistencies, the infoticle representation has an informational granularity of ±1.000.000 CHF (one infoticle too much or too less), for an averaged yearly budget of about 970.000.000CHF.

ACQUISITION	Third Party			
EXPENDITURE				
Departmental Category	2001	2002	change	number of infoticles
CONSTRUCTION & GEOMATICS	10.349	10.699	350	1
ENGINEERING SCIENCES	37.106	33.300	-3806	-4
NATURAL SCIENCES & MATHEMATICS.	33.086	36.202	3116	3
SYSTEM-ORIENTED SCIENCES	24.252	24.964	712	0
OTHER SCIENCES & SPORT	12.061	16.295	4234	4
CENTRAL SCIENTIFIC SERVICES	10.382	16.284	5902	6
ADMINISTRATION & INFRASTRUCTURE	3.632	1.788	-1844	-2

Table 4.7. Time alteration data (amounts in 1.000CHF)

4.4.4. Evaluation

The infoticle Electron World offers an alternative and truly dynamic representation of time-varying datasets that denote some kind of *flow*. Switching data dimension-representing force sets is a trivial solution to visualize multi-dimensional datasets while retaining a mono-dimensional representation. Instead of showing simultaneous multi-dimensional data dependencies that are difficult to interpret, the Electron World offers the more interactive and playful options of rapid dimension switching and subtle timeline manipulation. In this way, users continuously have full control over the conceptual connections they want to investigate, and can replay any animation sequence they like in different speed modes.

The long spline infoticle representation resembles and yet enhances that of traditional lines. In effect, the long infoticles connect the forces, and thus the data dimensions, in space, but are intrinsically capable to change their end points dynamically and smoothly, so that, in addition, dimensionality alterations can be continuously followed. Furthermore, the motion aspect enables users to immediately and cognitively perceive data evolutions in time, instead of having to compare static or numerical representations. The major advantage of the Electron World representation in comparison with most traditional approaches is its capability to merge *all* data dimensions and time-varying data values within a *single* interactive representation. Instead of switching static representations and comparing numerical tables or bar chart sizes, the flow of data-driven particles allows for successive analyses of differences in dimensions or time evolutions within the same contextual virtual world.

The Electron representation is not very effective in conveying measurable quantities of money flowing from and to the different data dimensionalities, but rather gives an intuitive estimation of *evolution*. Instead of providing users with exact numbers, the Electron World represents comparable bunches of connecting lines in time that need little quantifiable interpretation. At the same time, because the time-varying alterations cause only small quantities of infoticles to change, they can be more easily quantitatively compared. In addition, the volumetric sizes of infoticle clusters and the proportion of colors within these (and thus the acquisition dimension) these can be effectively detected and compared.

The Electron World is primarily based upon a *collective* infoticle analysis, as individual infoticle behavior is irrelevant for this visualization scenario. Instead of generating easily distinguishable motion typologies, all interface alterations result in common changes of infoticle groups that only differ in direction, volume and quantity. Similarly to the other visualization scenarios, the Electron World facilitates the perception of odd and outlying data patterns, as very small changes of a handful of infoticles at timeline alterations are more easily distinguishable than large-scale migrations. This feature is especially useful for visualizing time-varying financial datasets that, due to their large number of numerical values, become difficult to comprehend or compare by most traditional visualization means.

4.5. Boid

The previously mentioned infoticle visualization scenarios primarily rely on a simple set of external rules that determine the individual spatial behavior of each infoticle from a *holistic* view. These conditions specify spatial reactions based upon the time-varying update of data objects, but never consider the situation of the rest of the dataset during the evaluation and interpretation process. However, most information visualizations represent data tendencies of individual entries in comparison with simultaneous trends occurring in the whole dataset, so that specific global dissimilarities can be detected. For this reason, *self-organizing principles* have the intrinsic potential of representing data spatially in a meaningful way, as this concept bases its reactions mainly upon contextual and environmental parameters. In some other disciplines, self-organization and artificial life methods have been successfully used to mimic semi-intelligent behavior. Accordingly, the boid infoticle method attempts to embed organizational reasoning within the representation itself, instead of governing it externally like in most traditional approaches.

4.5.1. Goal

Many time-varying datasets consist of relatively large amounts of data entries that change in an extremely rapid and chaotic way. Visualizing such datasets is not trivial, and various information visualization approaches today attempt to present the dynamic trends within such datasets by generating static representations. For instance, stock market quotes are fundamentally unpredictable and disorganized, although some sophisticated statistical analyses suggest the existence of intrinsic relationships between specific corporate areas and related time-varying trends. Exposing data-driven particles to the evolution of stock market quotes would prove the validity and robustness of using motion characteristics and local behavior rules for valuable and complex information visualization purposes. Therefore, the next dataset consists of the historical daily stock market quotes of 500 different companies traded on the New York Stock Exchange. Notably, the fact whether such data is communicated to the application in real time or rather consist of a real-world simulation of a collection of historical quotes, does not play a relevant role.

Finding informational similarities within such large and almost chaotic dataset requires a complex comparison process, which often is executed 'off-line' in a dedicated calculation time slot well before the visualization initialization. In contrast, boid infoticles detect their informationally similar *mates* during execution time and then cluster accordingly.

Because of the continuous attraction and repulsion impulses of ever-changing groups of mates, a boid infoticle representation is in a continuous state of instability. This stress level is required to facilitate rapid behavior changes in reaction to data updates, but at the same time might lead to chaotic and thus complex behavior that is difficult to interpret. A specific balance has to be reached between slow and stable *equilibrium* and fast and unpredictable *chaos*, so that the representation has the capability to adapt to the relatively fast data value changes. Relevant parameters in this process are the duration of the application timeframe and the various boid variables that determine the overall flocking consistency: boids that cluster too slowly might lead to visual misinterpretations, whereas fast grouping infoticles might move too drastically to visually denote any informational meaning. In effect, these influencing factors need to be exactly determined during an empirical process of trial-and-error, as to date no algorithms exist that predict the emergent behavior of 500 internally continuously interacting elements.

4.5.2. Implementation

The following implementation issues had to be addressed in the Boid scenario.

- **Scenario Development.** The infoticle Boid scenario is limited to the presentation of update trends within an extremely dynamic and almost chaotic dataset. Instead of representing exact data values, the *way* the data values *evolve* is visualized. Because stock market trends are generally caused by external factors, the exact data pattern occurrences need to be explored through the use of alternative information sources.
- **Data Acquisition & Filtering.** The dataset consists of historical stock market quotes of one year, totaling about (±500 companies x ±200 working days) 12.631 data entries. It is acquired from a public website that accumulates the historical stock market prices of the 500 Standard & Poor's Index Directory (SWCP, 2003). This index consists of a representative sample of 500 leading companies in the most important industries of the U.S. economy. Although the S&P 500 focuses on the large-cap segment of the market, with over 80% coverage of U.S. equities, it is an ideal proxy for the total market. The dataset accumulates daily opening, closing, high and low prices and the corresponding trade volumes.
- **Data Abstraction & Conversion.** As the dataset is relatively simple in nature and the data identifiers are depicted by moderately short text labels denoting the symbols of the individual companies, abstracting and converting the data is straightforward.
- **Metaphor Adaptation.** The metaphor is based upon the infoticle boid algorithms as explained in detail in Section 3.5.4. Infoticle – Simulation – Behavior.
  - **Emitter.** The emitter creates all infoticles at once, and plays no further significant role in the visualization simulation.
  - **Data Similarity.** Companies with equal *relative* stock market price *changes* (measured as a percentage of their closing price) are attracted by the data similarity clustering tendency. In practice, this means that companies with *parallel* closing price line diagrams are spatially clustered for that period in time.
  - **Infoticle.** Each infoticle represents a single company. As some companies are added to or removed from the Standard & Poor's Index Directory, these infoticles have a static stock market quote and are simulated accordingly.
  - **Tools.** To facilitate the comprehension of the resulting visualization, tools were limited to shapes and traces. The scene does not include any forces or filters as these would inevitably influence the fragile balance of internally related dependencies.
- **Application Reconfiguration.** The main reconfiguration task deals with determining the exact values of the boid simulation algorithm variables that produce a desirable flocking simulation. This process is empirical in nature, as the extremely rapid update pace of the data objects stresses the normal flocking behavior considerably and the outcome cannot be foreseen. Boid behavior variables influence the required adaptation speed to reach a true representation and a perceivable motion coherence. These

parameters depend on the desired dynamic flock behavior that is required to adapt to each new data update within the predefined application timeframe. Furthermore, boid infoticles should never move to positions too far away due to low centering forces or be stuck in fixed spatial positions due to high collision avoidance forces.

4.5.3. Result

The main design concern is related to the exact numerical determination of the most suited flocking behavior variables, due to the unpredictable, self-organizing nature of the boid infoticle methodology. As global flocks typically move constantly, and do not stay fixed around the same position, one other implementation matter dealt with the relative motion of the whole flock itself, as it typically tends to move outside the active view frame.

Please note when investigating the stock market bar charts that boid infoticles cluster only when the *relative change* of stock market quote is similar, not the stock market price value. Therefore, infoticle boids cluster when the respective lines within a typical stock chart are *parallel*, not necessarily when these are graphically close together or overlap.

Figure 4.27 shows the capability of infoticle boids to denote individual exceptional behaviors within a large collection of time-varying data entries. As the corresponding stock quote chart proves, those data-driven particles that cause the shape to separate or bulge considerably, have experienced a stock market evolution that is significantly different from the dataset majority during that specific database timeframe. More specifically, within the bulge generated by WIN and COST, one can perceive that COST is returning to the main flock, as its exceptional price fall ended some time ago and it started to behave similarly to the imaginative *flock average*, the S&P500. One can also perceive a gaining (bottom) and loosing (top) area within the shape. Notable, some other significant bulges are present on the same boid shape, as boid visualization artifacts typically have to be considered from many view points and directions due to their three-dimensionality.

Similarly, Figure 4.28 clarifies the directional clustering and shape separation after significant data discrepancies of small amounts of data. More specifically, a single blob shape, separated from the main flock, contains five different companies with equal data evolutions. Figure 4.29 shows the historical trajectories of three boid infoticles in relation to the main flock, demonstrating the continuous dynamic nature of the boid visualization technique. In fact, these trajectory lines prove that all boids move and jigger constantly between specific thresholds, and continuously adapt to the ever-changing situation surrounding them. Because they do this collectively while being grouped, this behavior is not visually confusing nor disturbs the accurate perception of significant events. This figure also illustrates how exceptional stock market price alterations dramatically alter the relative position within the flock. After a relatively chaotic initialization period, most boid infoticles generate interpretable trajectories. IBM follows the S&P500 index closely, and therefore has a smooth and regular trajectory. WIN reclustered with IBM during periods of similar price evolutions, and moved far away apart when not. Note that Figure 4.29 is a two-dimensional perspective image of a three-dimensional world, so some visual occurrences might look less significant than they actually are, and occlusions might disturb a correct interpretation.

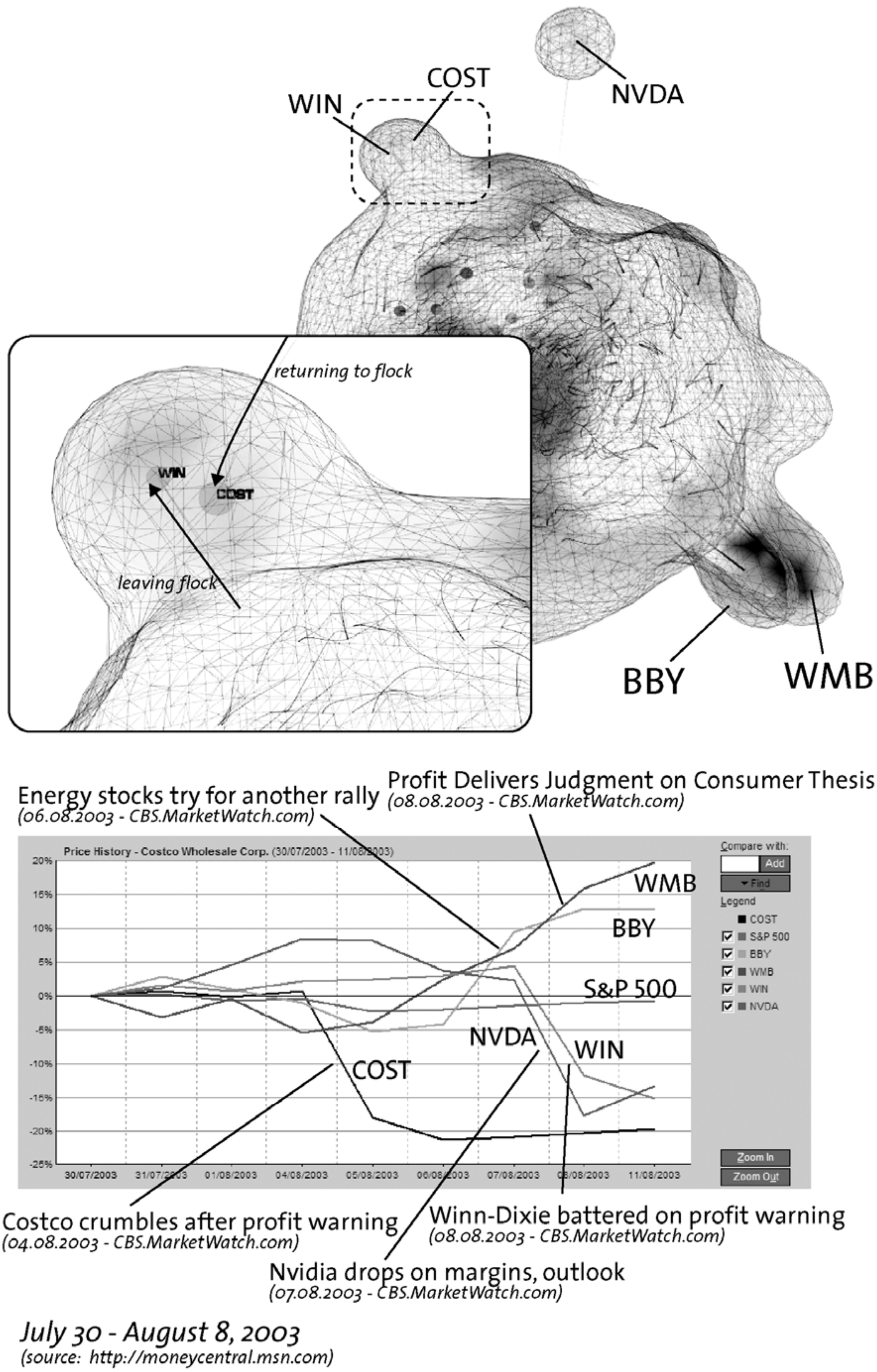


Figure 4.27. Boid shape interpretation.

.

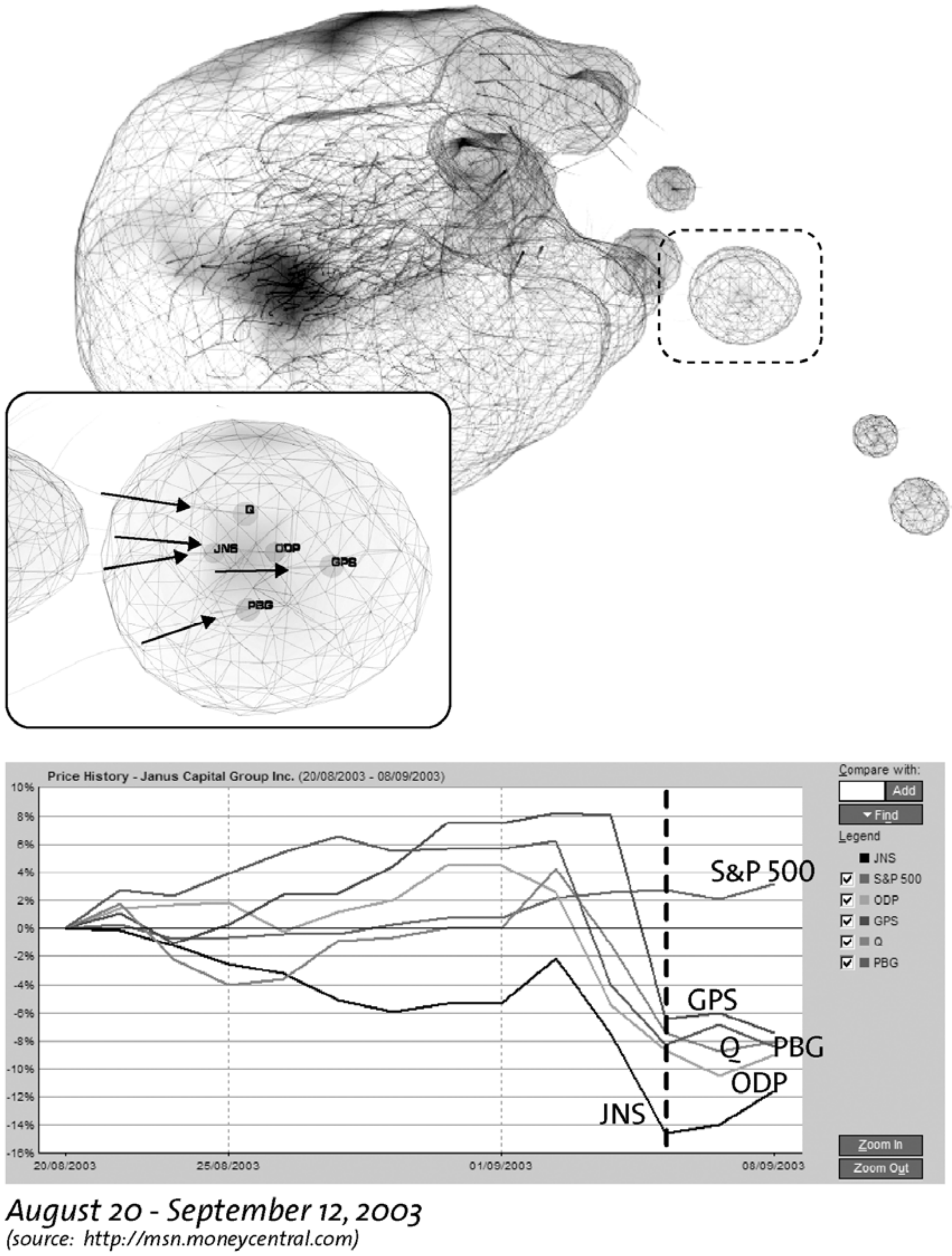


Figure 4.28. Boid clustering.

Next to the individual behaviors mentioned above, boid infoticles have the intrinsic capability to *behave collectively* meaningfully in relation to time-varying evolutions. Figure 4.30 shows how two very stable global behaviors (on August 21 and September 2, 2003), in which almost all boids are directed equally, become disturbed by a sudden directional *implosion* of outlying boids towards the flock center on August 26. Interestingly enough, initial research attempted to explain the implosion phenomenon, only to detect little or nothing significantly within the dataset.

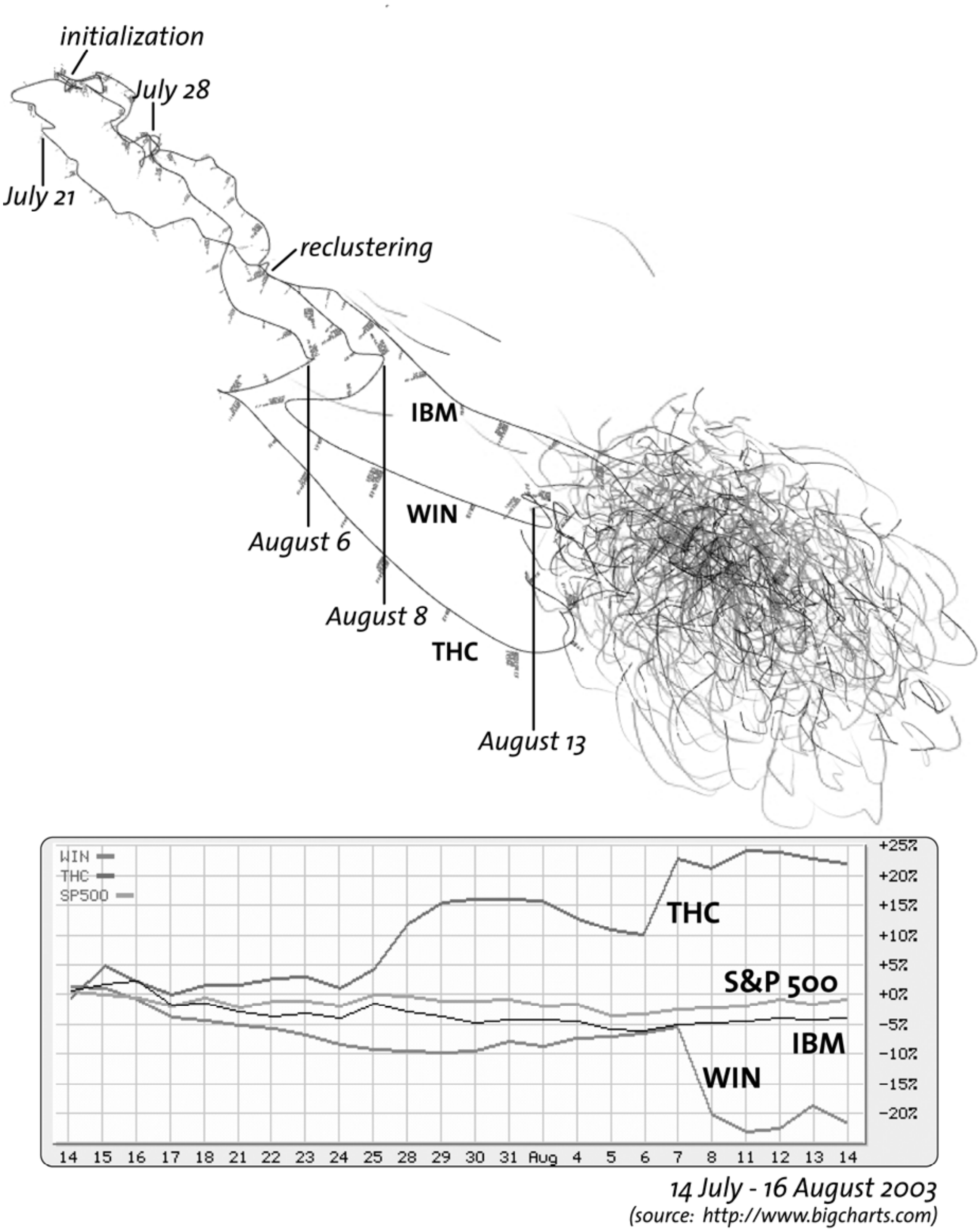
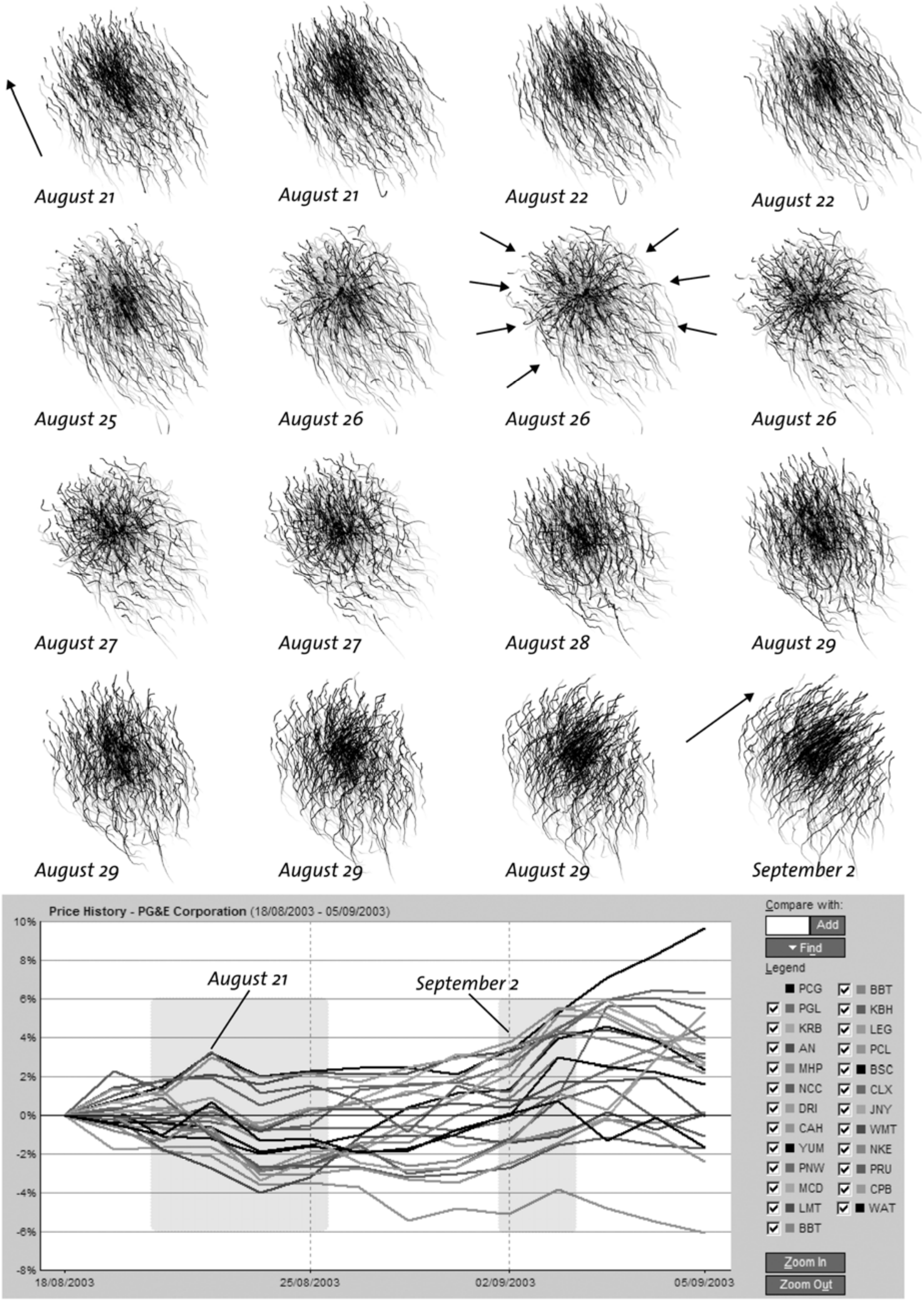


Figure 4.29. Relative boid behavior.

However, only further investigations revealed the implosion to be an instable state between two very distinct and stable evolutions: more than 50 companies at the outskirts of the flock were found to have undergone highly similar price quote evolutions, a collective data phenomenon that was not detected before. Possibly, nothing of note was happening with these companies on those days and the majority of stock prices were riding on external market variables, meaning that *market mood* is probably the strongest correlate during these specific time periods.



NYSE - August 16 - September 5, 2003
(source: http://moneycentral.msn.com)

Figure 4.30. Global boid behavior.

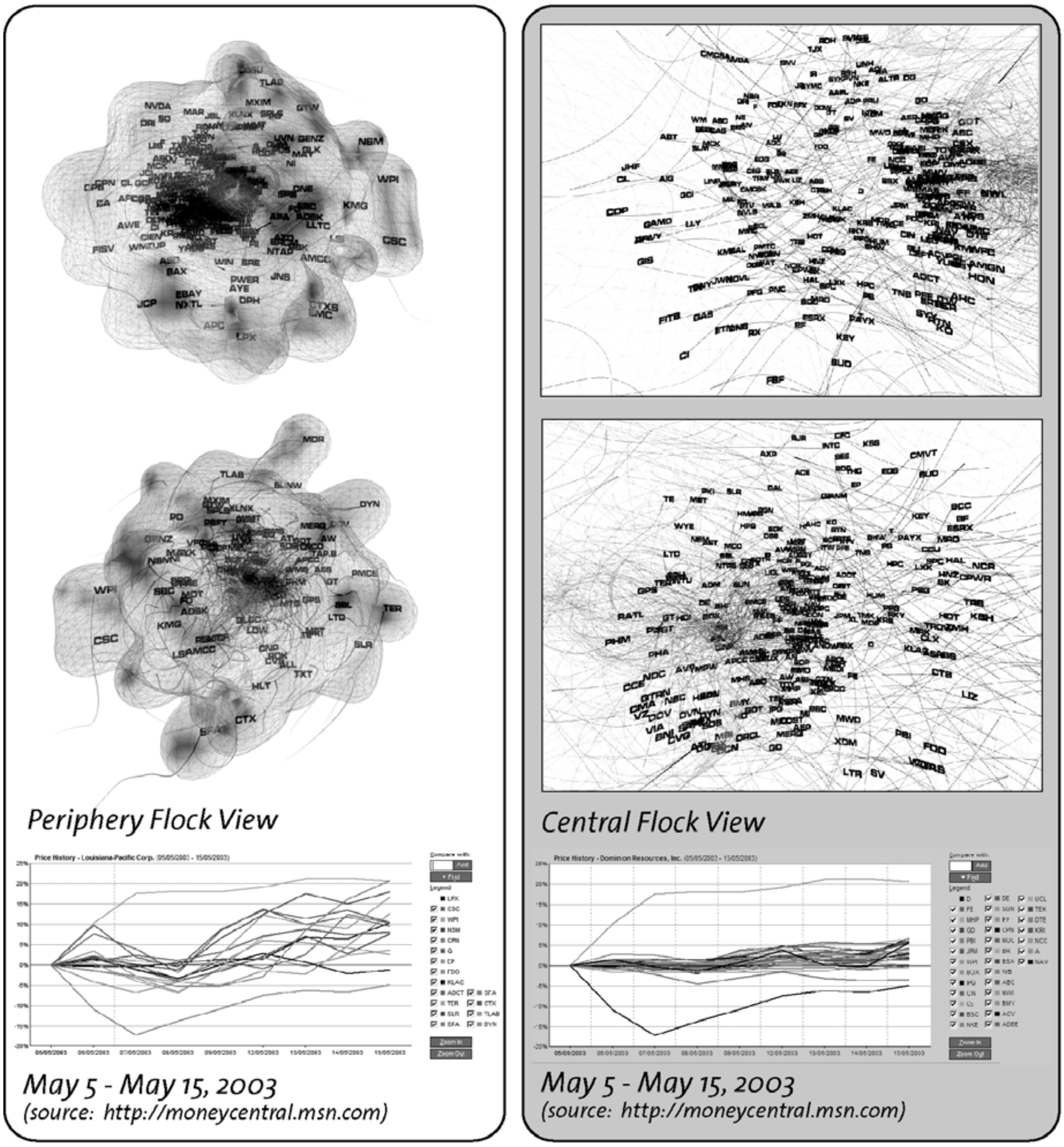


Figure 4.31. Spatial zoning.

Figure 4.31 demonstrates the *spatial zoning* tendency of boid flocks: the left column denotes those boid infoticles found in the *flock periphery*, in contrast to the right column, which investigates the boids detected at the shape center. Analysis of the corresponding stock market price evolutions clearly shows that data objects with relative large data value changes are positioned at the flock outskirts, whereas similarly and steadily changing entities remain closely together around the main focal point.

Whereas the previous images denote infoticle boids with behaviors that resemble fish or bird flocking, Figure 4.32 and Figure 4.33 illustrate what occurs when both flock centering and collision avoidance correspond to considerable high values. As a result, the boids either act as atomic entities within individual spatial borders (Figure 4.32) or behave like swirling electrons, as depicted in Figure 4.33. All boid images demonstrate that boids use spatial and directional *expulsion* as a behavioral technique to correlate unique and outlying time-varying patterns within the dataset.

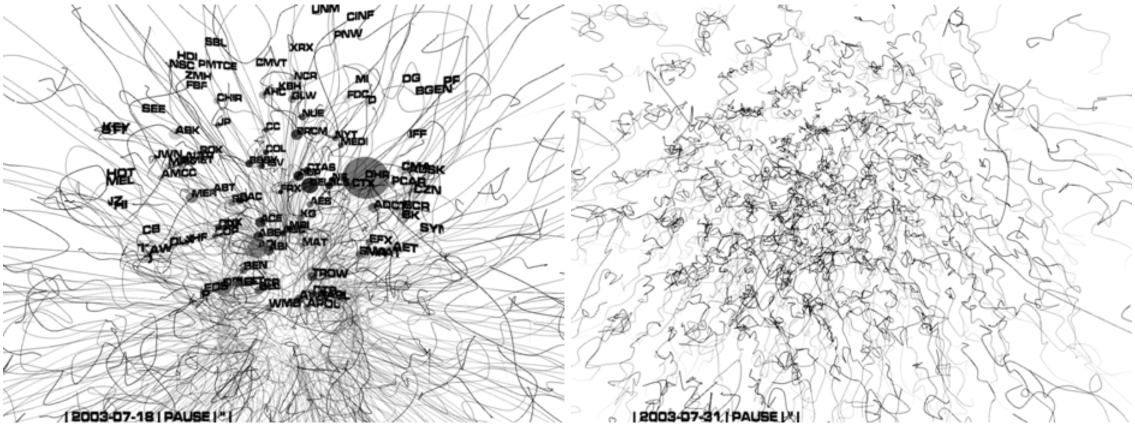


Figure 4.32. Collision avoidance influence.

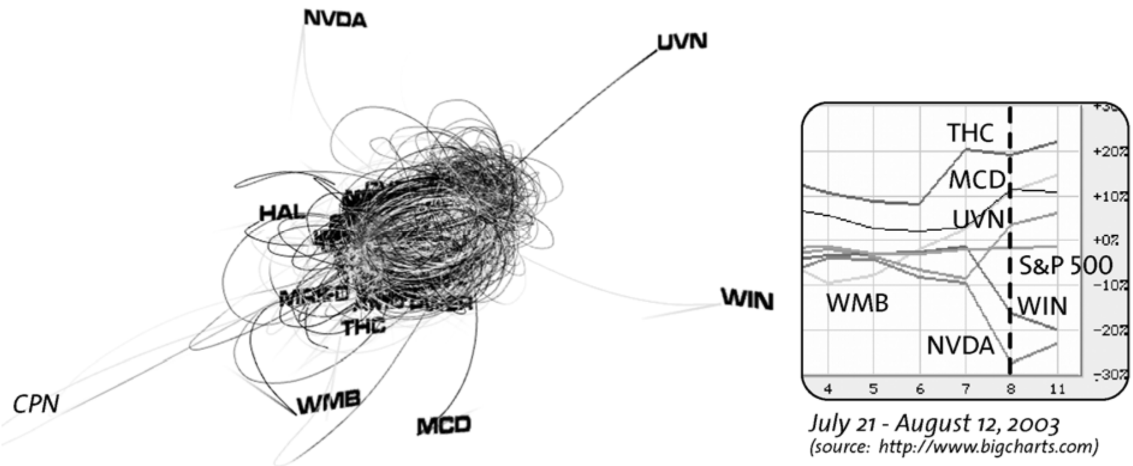


Figure 4.33. Flock centering influence.

4.5.4. Evaluation

The Boid infoticle method offers an alternative method to visualize rapidly changing datasets in a visually dynamic, but stable way. In fact, the boid infoticle algorithms extend an already known, useful visualization method for static data (Proctor and Winter, 1998), and has proven its usefulness for complex, time-varying datasets. Its effects are unpredictable in nature, but result in recognizable and interpretable data patterns. In contrast to the Galaxy World, the majority of the infoticles shows no distinguishable individual behavior, but instead behaves similarly, closely together in *flocks*. Only data outliers or significant trends cause this continuous behavior consistency to break, requiring a limited amount of visual attention from users to recognize these unexpected events.

The boid method is a valuable approach for visualizing complex datasets, certainly for users without any prior knowledge about the specific time-varying characteristics. One can imagine many other datasets that consist of large amounts of unpredictable numerical values that might show some time-varying similarities. For instance, datasets collecting numerical measurements of sounds, vibrations and other sensorial data from a physical engine prototype simulation might be simultaneously compared and analyzed with boid infoticles to detect time-varying oddities or global evolutionary similarities in any of the measured dimensions.

The emergent data clustering is caused by local interactions within the information representation itself, and not by pre-calculated data dependencies or spatial positions. Accordingly, several data patterns were detected that had not been predicted beforehand. In fact, one can challenge the theoretical or practical possibility to detect some boid patterns for 500 companies with other traditional non-statistical visualization methods. Because of the unpredictability of the boid simulation method, the exact reproducibility of data patterns is not guaranteed when the infoticle positions and directions are not exactly equal during the application initialization. However, the robustness of the clustering algorithms will produce practically similar patterns, although some slight spatial differences might occur.

The main remaining issue of this visualization method relates to the correct interpretation of the perceived patterns: although global behavior changes clearly denote the significance of simultaneously occurring tendencies within the dataset, they do not clarify the exact reasons that caused them. Typically, detecting the hidden driving forces behind such unexpected data patterns requires alternative means of analysis and a well-defined limited data pattern timeframe in which the event took place. Whereas individual infoticle *expulsions* generally denote specific exceptional changes in the stock quotes of the corresponding corporations during that period of time, *collective* behaviors are more difficult to interpret, and typically require slightly more observation time to be effectively perceived.

Time-varying datasets can be visualized with different time granularities by simply adapting the database timeframe duration. However, boid simulations themselves cannot be easily sped up, as the self-organizing nature of the method requires a specific amount of adaptation time after each introduced data update and behavior instability. Most importantly, the visualization scenario is able to animate only a limited amount of boids, as the many spatial and data interdependencies need to be continuously checked and thus require a considerable amount of computing performance.

5. Analysis

Information visualization supports users in building up a mental data model by artificially constructing a corresponding visual representation. An effective visualization metaphor facilitates a rapid understanding of this model, and allows for a direct representation manipulation for exploration purposes. Such a metaphor requires specific data evaluation principles, which generate abstract connotations between visual patterns and meaningful data structures. As a result, such rules build up an elementary visual *grammar* as an empirically defined base for interpreting emergent pattern formations.

This chapter takes a holistic view on the data-driven particle method, and analyzes its potential for information visualization. The applications described in Chapter 4 function as real-world demonstrations of conceptually distinct infoticle metaphor approaches, and are comparatively analyzed in relation to their fundamental visual cues. As a result, this chapter will describe the emerging grammatical principles and semantic rules, such as the historical data evolution clustering of the *equilibrium adaptation timeframe*. Furthermore, this chapter investigates the possible usages, qualities, shortcomings, ideal dataset characteristics and usability issues relating to the data-driven particle visualization methodology.

The following chapter is used to justify specific design choices in the context of time-varying information visualization and to gain insights with an eye on extending the visualization method towards alternative dataset usages or presentation media. It explains potential users the various qualities and problems that relate to the methodology and, as such, can serve as an empirically defined basis for other visualization metaphor designers interested in time-varying data phenomena. By relating this novel visualization metaphor to the following analysis framework as an integral part of the presentation, a conceptual model emerges that can be employed for future work. Ultimately, several core design principles are derived that are valid for more general time-varying visualization approaches.

5.1. Grammar

A specific *grammar* constitutes a set of rules for combining the words used in a given language. Similar conceptual principles can be empirically observed that govern an infoticle representation, and can be derived out of the core spatial organization principles described in the previous chapters. In addition, this section analyzes how some of the emergent data-driven particle phenomena can be exploited for different data exploration goals.

Because of the novelty of motion utilization for information displaying purposes, and of dynamic infoticle patterns in particular, it is useful to investigate the effect of the rules that drive the representation. Therefore, this section explains the informational values of the visual cues that appear during an infoticle simulation. Subsequently, potential users can look up and learn the infoticle grammar to comprehend its inner workings and understand the meaning of the visual patterns. In effect, the following paragraphs describe a new *visual language* that is built up by individual and global data-driven particle behavior.

5.1.1. Influences

The infoticle method is not specifically suited to represent *exact* data values, as the representation is predominantly generated by dynamic update characteristics. Figure 5.1 shows that, following the infoticle methodology, infoticle behavior is primarily determined by the following three criteria.

- **Data Value.** The exact data object values determine the *relative spatial location* within the spatial scene, because the infoticle either is attracted or repulsed by a set of corresponding infoticle tools, or attempts to be near similar flock mates.
- **Data Value Change.** The relative data value change in comparison with previous historical values considerably influences the gradual *evolution of spatial behaviors*. For instance, depending on the comparative change after a data update, a star infoticle can only be followed by an electron or a comet pattern.
- **Data Update Frequency.** Complementary to the data value change, the data update frequency also determines some infoticle behavior characteristics. A specific fast sequence of relatively small data value changes (e.g. noise) might not alter the data pattern typology but some of its more *detailed visual features*. For instance, a comet infoticle that is updated to equal data values (so no data change but only an update) will stay in a comet state, but might exhibit a larger orbit path. Also, fast data update frequencies typically produce more irregular infoticle behaviors and tracing artifact shapes in comparison to slow updates.

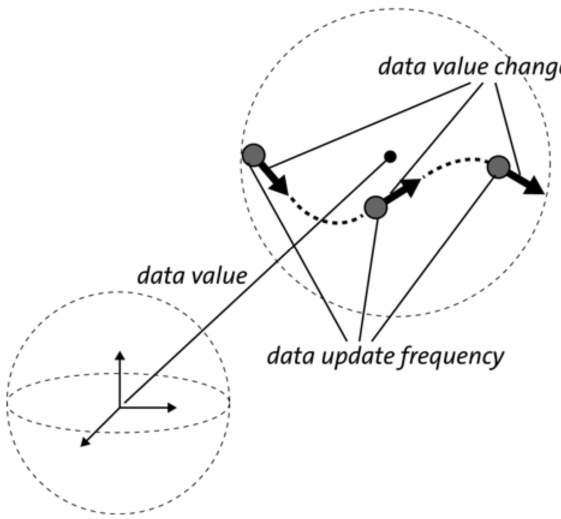


Figure 5.1. Infoticle pattern influences.

The combination of data update characteristics and exact or relative data value changes thus governs the visual dynamic outcome and the spatial position of an infoticle. This behavior is determined by two successive influences, applied in parallel to the whole infoticle collection.

- **Internal.** All infoticles, even when originating from different emitters, are controlled by an identical internal rule set. These rules are able to alter any infoticle attribute, producing specific visually recognizable effects, and are processed immediately after a data update for all infoticles, although, for performance reasons, some preferential ordering can occur (see Section 3.7.3. Infoticle – Implementation – Data Processing). This optimization method shifts specific infoticle alterations at slightly different points in time, although these are perceived as occurring simultaneously. Notably, these internal behavior mechanisms are hidden from users so that their outcomes typically need some explanation or metaphor manipulation to be fully comprehended. Two fundamentally different sorts of internal rules are identified, of which only one is applied during any infoticle visualization.
  - **Independent.** These behavior rules are applied holistically, independently of the contextual environment and thus are solely based upon relative data value changes. As a result, one is able to intuitively compare subtle variations in visual patterns of data items that have been subjected to *similar data histories*. Because of the lack of contextual information, different types of data histories build up dissimilar spatial behaviors and spatial trajectories. Subsequently, independent behavior rules articulate distinct formal typologies that are easy to distinguish by the human eye.
  - **Environmental.** Infoticles guided by local behavior rules consider the state of the surrounding infoticles for determining their own attribute alterations. These intrinsically contradicting influences typically result in fine, smooth behaviors that are more difficult to discriminate. Infoticles that do catch the eye, therefore, mostly have been subjected to update histories that are significantly different from the rest. In this sense, the infoticle system is able to filter out odd update tendencies by a dynamic process of *self-organization*.

- **External.** Next to the internal behavior rules reacting immediately upon data updates, infoticle attributes are altered by external influences that are invoked depending on spatial distances from the tools within the scene.
  - **Attraction.** Point forces gradually attract all infoticles with equal data values. The immediate infoticle *redirection* towards the new average force center after a data update enhances the perception of the resulting visual pattern considerably. First, each updated infoticle is rotated directly towards the active force, as otherwise it would need a very long adaptation time span to finally reach a force far away. The *orbit* rule avoids that an infoticle passes by the force because of a relatively fast speed accumulation. In that respect, these particular behavior rules are short, sudden and easily comprehensible influences that increase the force attraction effect considerably.
  - **Repulsion.** Filters bounce off all infoticles with dissimilar data values, creating spatial regions of related information. As users are more familiar with everyday attracting forces such as gravitation and magnetism, the infoticle metaphor does not introduce invisible, abstract repulsion influences, but uses clearly recognizable boundary surfaces instead. Consequently, the resulting visual effects are easily identifiable, so that no other fine-tuning behavior enhancements are needed. However, filter-collided infoticles might still be attracted to the same force, and therefore get trapped in a confusing perpetual effect of filter repulsion and force attraction. For this reason, a continuous *infoticle state detection* is implemented, which blocks subsequent further attraction by the same force for infoticles that have recently bounced into a filter.

5.1.2. Dynamic

An infoticle representation can be perceived and analyzed in contextually different situations, each of which presents distinct visual patterns and thus offers a distinct conceptual dataset interpretation. The infoticle metaphor is primarily meant to be perceived dynamically, as it represents the continuous data updates at a faster pace than the originally stored database timeline. This dynamic time simulation also enables a direct interaction with the evolving representation at different moments in time.

Generally, an infoticle representation can be dynamically analyzed in a similar way as scientific particle visualizations of physical phenomena, such as smoke dispersion or temperature distributions. Differences in cluster densities can be easily recognized, and are related to the size and amount of infoticles with similar data values. Different data update typologies, such as fast, chaotic, linear, sudden, etc. can be cognitively mapped onto infoticle behaviors. The data-driven particles are adapted directly and in real time, and the resulting influences are shown incrementally, offering users a smooth perceivable transition between distinct states. In particular, the following dynamic perception issues are recognized for each of the application prototypes.

- **Modeling.** The dynamic outcome of the modeling world, and thus the meaning of the representation, is mainly influenced by user interaction. Users are able to perceive the influence of force or filter alterations on the speed and direction changes of infoticle clusters that flow and regroup. For instance, when the position of a single departmental force is altered in the scene, the specific infoticles that suddenly alter direction and the relative speed with which this change takes place can be visually tracked.
- **Galaxy.** The Galaxy World relies extensively on the effective perception of dynamic infoticle behaviors. In fact, the infoticle behavior rules inherently determine a specific pattern emergence sequence, following predetermined time-varying evolutions. Table

5.1 illustrates how the different Galaxy patterns interrelate when the system is in a dynamic state. For instance, it shows how each comet and electron pattern evolutionarily derives from the star pattern after a specific time span. A comet can be perceived by spotting a sudden event, while electrons are formed through a slow, incremental process. The star pattern always depends on the transfer phenomenon, and is triggered by a specific force distance condition. This table also indicates whether specific patterns can be detected by considering *individual* (comet, quark) or *global* (star, electron, burst) infoticle behaviors during sudden or long-term events.

Pattern	Infoticle Quantity	Pattern Before	Trigger Action	Start Feature	End Pattern	Other Visual Features
Transfer	single	any	sudden event → *directional change*	any	any	straight line
Global	multiple	stars comets electrons	slow evolution	high density	diffusion	stable density parallel directions
Time	multiple	any	sudden event	any	any	directional change color change
Star	single	transfer	force distance → *orbit*	stable circular trajectory	transfer electron comet quark	slow speed
Electron	single	star	slow evolution →*drag* → *force distance*	fast spinning	comet quark transfer	circular trajectory spinning high speed
Comet	single	star electron	sudden event →*speed increase*	stable elliptical trajectory	any	high speed
Burst	multiple	multiple transfers	sudden event	equal directions	any	same speed
Quark	single	sequential transfers	multiple sudden events	chaotic behavior	star	high speed

Table 5.1. Galaxy pattern evolution.

Accordingly, Figure 5.2 illustrates the time-dependent and evolutionary relationships between behavioral patterns for a dynamic Galaxy World.

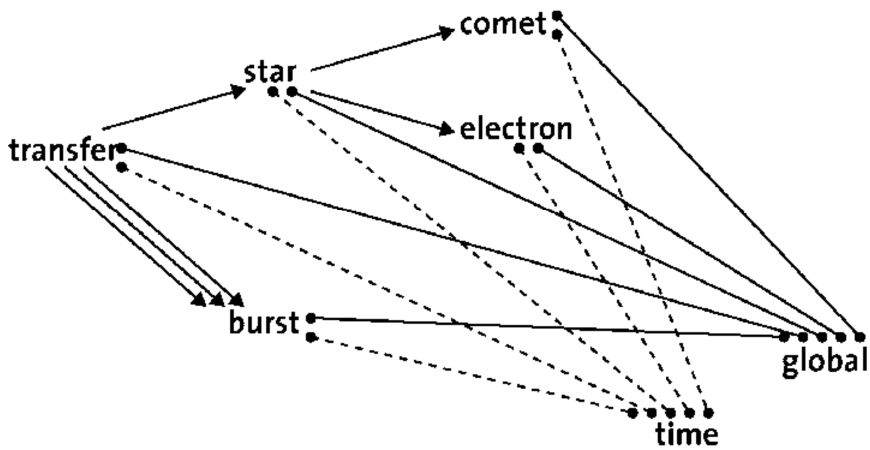


Figure 5.2. Galaxy pattern evolution.

Notably, individual infoticles that behave similarly, such as electrons or stars that orbit the same force, are perceived in group instead of atomically, because of the visual clustering effect of equal motion typologies. This perceptual phenomenon decreases the amount of attention needed to follow the thousands of moving points within the scene, so that infoticles in transit from one pattern to another are easier recognizable by human cognition. Motion pattern types are fundamentally different in their speed, shape, time-dependency and infoticle quantity, and generate abrupt dynamic changes that the human eye can effectively distinguish.

- **Electrons.** The Electron World presentation relies solely on user induced data dimension switches, and time direction and timeframe duration changes. Instead of comparing static representations at different moments in time, data alterations are immediately and smoothly visualized when the timeline is animated. In practice, infoticles with unchanged data values remain clustered around the same forces, whereas only those infoticles that are subjected to some form of *data value delta* switch from one cluster to another, and thus can be instantly recognized.
- **Boid.** The infoticle boid method differs considerably from the previous approaches as no clearly defined visual patterns can be recognized. In effect, boids continuously generate smooth animations and morph between different dynamic phases instead of displaying distinct motion typologies. By consequence, boid alterations are more difficult to perceive, and mostly require interpretation on a holistic level, as the patterns emerge out of a continuous process of numerous local interactions.

  Boid World pattern recognition is partly based upon an uninterrupted comparison of animated clusters and blob shapes generated by grouped infoticles. Infoticles that are directed away from the center or are positioned on the tail or head of the flock have experienced extraordinary data histories in comparison with those that are in the center. As static boid shape interpretation is primarily based upon formal analysis, subtle shape alterations are only identifiable by continuously tracking *shape morphing*. In addition, even the dynamic behavior of boids can be categorized, as specific information visualization research has already classified flock behaviors in comparison with natural phenomena (Kadrovach and Lamont, 2002). As Figure 5.3 indicates, numerous types of swarm formations exist that could be used for meaningful visualization purposes. The vertical axis represents global or regional influences, the lateral axis denotes the degree of order (e.g. neatly ordered as fish or chaotic like insects), whereas the depth axis represents the degree of environmental information sharing through some form of communication.

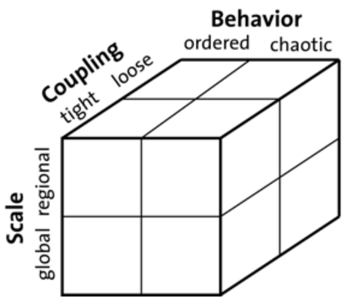


Figure 5.3. Boid behavior classification (Kadrovach and Lamont, 2002).

5.1.3. Static

An infoticle representation can also be analyzed in a *static* state, hereby conveying fundamentally different data patterns than those mentioned in the previously mentioned dynamical analysis. Generally, a *frozen* infoticle world is visually searched for infoticle clusters that are close to one another or are heading in an almost identical spatial direction, as such patterns typically denote a certain degree of data similarity. Especially the Galaxy and Boid World scenarios are capable to represent different time-varying dataset relationships statically.

- **Galaxy.** A static Galaxy representation can be analyzed in two fundamentally different ways.
  - **Spatiality.** Figure 5.4 shows how most dynamic patterns emerge within specific spatial zones in relation to the visualization center and the average vector position. Users are able to focus on these well-defined regions when searching for particular document usages. Table 5.2 demonstrates, for instance, that comets are generally found at the outskirts of the visualization, comets and electrons around forces, and quarks in the center of the scene.

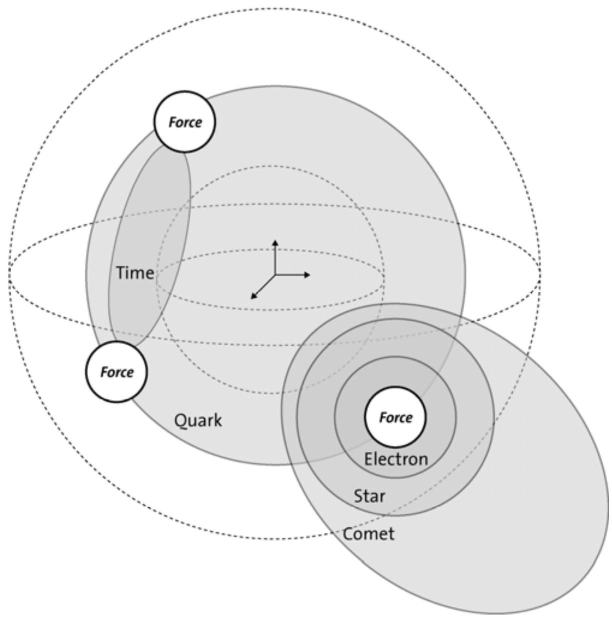


Figure 5.4. Galaxy pattern spatiality.

Pattern	Spatiality	Distance (d)
Transfer	between forces	towards force
Global	between forces	average all forces < d < average catch range forces
Time	anywhere	-
Star	individual force vicinity	center of force < d < force catch range
Electron	individual force close vicinity	center of force < d < collision avoidance range
Comet	stable elliptical trajectory	force catch range <d < visualization outskirts
Burst	anywhere	-
Quark	in between forces	visualization center < d < average catch range forces

Table 5.2. Galaxy pattern spatiality.

– **Formality.** The interpretation of dynamic infoticle behaviors is partly based upon the continuous tracking of infoticle trajectories, forcing users to be attentive during relatively long time spans. In addition, infoticles that are far apart are difficult to compare, and certain pattern evolutions can be easily missed when the focus of attention has shifted to other or unrelated events. In contrast, the static shapes generated by the trace ribbons enable users to cognitively interpret formality, spatial area and general direction in relation to data update characteristics. Table 5.3 explains the visual attributes of timeline ribbons for all Galaxy World patterns.

Pattern	Trace Formality	Other	Evolution (Δt)
Transfer	*straight direction*	*towards force*	*- evolves into other pattern close by force*
Global	*- any forms - mostly stars & electrons*	*dense spatial cluster*	*density changes*
Time	*- changes at equal points in space & time - mostly consist of transfers*	*equal color intensity*	*diffuse (all) into different patterns*
Star	*stable circular trajectory*	*around force*	*radius decreases*
Electron	*stable small circular trajectory*	*around force*	*many orbits no radius change*
Comet	*stable elliptical trajectory*	*around force*	*radius decreases*
Burst	*straight, parallel lines*	*spatial cluster with equal color intensity*	*evolve (all) into equal pattern (star)*
Quark	*- chaotic trajectory - circular orbits & straight lines*	*towards & around multiple forces*	*quickly switches from transfers to stars*

Table 5.3. Galaxy static pattern detection.

– **Boid.** The blobby shapes spanning the spatial positions of the boids considerably reduce the perceptual effort required to recognize outlying infoticles and small infoticle clusters. Infoticles that lay outside of the flock center generate relatively large bulges on the blob surfaces, which are easier recognizable than tentatively measuring relative proximities to the estimated main flock center.

5.1.4. Semantics

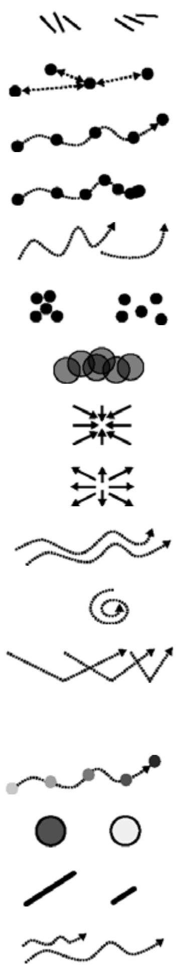
The infoticle metaphor's core visualization engine is based upon the combination of three behavioral concepts.

– **Similarity.** The infoticle data-mapping algorithms translate data value similarity into clustering or spatially proximate infoticles that possibly have equal directions or short trajectory parts. In addition, parallel data update characteristics result in similar trace ribbon formalities, even of distant infoticles, whereas identical data update histories ribbons are closely adjacent. Notably, these effects are not generated by singular Cartesian coordinate mapping mechanisms. Instead, positional evolutions driven by gradually escalating influences build up a globally valid environmental context.

– **Consistency.** Data similarity and data variation are conveyed by respectively maintaining and breaking the consistent evolution of individual dynamic behaviors. This effect is simultaneously mirrored by static representation artifacts produced by tracing the resulting spatial positions and directions. Data value and dynamic inconsistencies are mapped onto sudden behavioral alterations and corresponding graphical shape irregularities.

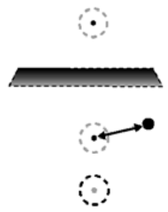
– **Causality.** Users need to understand the previously mentioned similarity and consistency methodologies to orientate themselves in both the spatial representation and the corresponding abstract dataset structure. Therefore, the cognitive interpretations of such patterns should be relatively similar to the dynamic update characteristics they represent. For instance, rapidly updated data entries require vivid representations and large value changes should be represented by similarly sized changes in their visual counterparts. In effect, the causality between data update features and their graphical consequences needs to be graspable by ordinary users.

Consequently, the infoticle metaphor transcends traditional coding mechanisms that translate data values into individual graphical cues in a singular (a specific data value is uniquely mapped onto one single specific visual cue) and bidirectional way (each visual cue corresponds to one specific data value). In contrast, many infoticle patterns represent informational values by a simultaneous *combination* of distinct graphical codes that can be perceived in different contextual situations. Table 5.4 demonstrates how the concept of data-driven particles takes advantage of merging basic perceptual syntax mechanisms to convey information, as it gives a detailed visual grammar analysis of the visual cues that build up the emergent infoticle patterns.



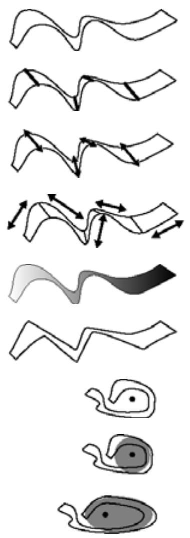
Visual Instantiation	Graphical Code	Semantics	Informational Value
	Clustering		
	directionality	*goal directed*	*immediate data value similarity*
	proximity	*groups of elements*	*data value similarity evolution*
	motion	*evolution*	*adaptation between data updates*
	drag	*resistance*	*time passed since last data update*
	behavior	*spatial alterations*	*dynamic data update typology*
	density	*volume size*	*degree of data similarity*
	blending	*overlapping*	*entity type quantity*
	converging	*flocking behavior*	*long term relative data dissimilarity*
	diverging	*flocking behavior*	*long term relative data dissimilarity*
	parallelism	*flocking behavior*	*long term relative data similarity*
	swirl	*motion typology*	*long term data similarity*
	collision	*opposite velocity*	*data dissimilarity*
	Infoticle		
	brightness	*numerical value*	*time passed since last data update*
	color	*entity type*	*fundamental difference in data attributes*
	length	*numerical value*	*infoticle speed*
	speed	*numerical value*	*time passed since update*

Tools



point	*influence*	*data force*
plane	*border*	*data filtering*
distance	*influence*	*behavior switch*
circular plane	*icon*	*selection range*

Ribbon



shape	*historical spatial path*	*dynamic data update typology*
irregularity	*change*	*exact data update*
width	*numerical value*	*parallel data frequency*
length	*time measurement*	*past timeframe*
gradient	*vector direction*	*history direction*
straightness	*connectivity*	*time-limited transfer between patterns*
orbit	*connectedness to center*	*force data value similarity*
circle	*stable connectedness*	*stable force data similarity connection*
ellipse	*repeated connectedness*	*extensive force data similarity connection*

Tools

stable	*stationary evolution*	*slow data update history*
chaotic	*moving evolution*	*rapid, considerable data update history*
size	*volume*	*degree of data similarity*
form	*cognitive interpretation*	*data update typology*
smoothness	*regularity*	*data similarity*
bulge	*irregularity*	*data dissimilarity*

Table 5.4. Infoticle pattern perceptual syntax.

5.1.5. Similarity

Analogous to most information visualizations, spatial infoticle-infoticle and infoticle-force proximity denotes a degree of *data similarity*. However, because the system evaluates data similarities in real time and the resulting visual patterns are dynamic in nature, determining a directly quantifiable distance-similarity relationship is often impossible. Instead, a certain degree of conceptual relationship is denoted by *relative proximity* (in relation to other infoticles nearby) and *directionality*, which is progressively enhanced during the database timeframe evolution (see also Section 5.1.9. Analysis – Grammar – Equilibrium). However, some precautions are called for when interpreting infoticle density clusters.

Foremost, data similarities are not only represented by infoticles in close *proximity*, but also by infoticles with similar spatial *directions* or equal dynamic *motion* patterns. In fact, cognitive science has demonstrated that dynamic points with similar directionality or motion typology are visually clustered due to the Gestalt law of Common Fate.

- **Galaxy.** Because of the evolutionary nature of the dynamic simulation, proximity does not denote resemblance in an exact mathematical way. Consequently, precise distances and data similarities do not relate directly, and thus cannot be quantifiably measured nor compared. In practice, this means that infoticles at slightly different distances from forces do not denote increasing data value dissimilarities, because the attraction force strength is not dependent on the data value and the subjected infoticles might have arrived at different points in time. Instead, the distance force-infoticle represents an identical data value relationship measured in time.
  - **Force.** Force-grouped infoticles such as stars, electrons and comets, contain *equal* data values without any gradual similarity level. Distance is related to the time passed since this relationship was accomplished and the frequency of this relationship.
  - **Time.** Infoticles within time clusters have similar *data update histories*, as they have been influenced by similar force sets. In addition, adjacent infoticles within these groups orbit proximate average forces, and thus represent similar *data value histories*.
  - **Burst.** Burst infoticles represent *contextual* data similarity, as the corresponding information typically is accessed simultaneously by the same user or region. Such clusters represent either some dependence in informational content or might be related technically on the level of file typology.
- **Boid.** The boid world is generated by internal, flock-mimicking influences that prefer data similarity between neighboring mates, so that inter-infoticle distances have mathematical connotations. However, users should be continuously aware that the system requires a specific time span (see Section 5.1.9. Analysis – Grammar – Equilibrium) to represent this true state, as the infoticles still might need to overcome specific spatial distances to reach their favorite neighboring mates. Consequently, a boid representation is best analyzed at the end of an application timeframe.

5.1.6. Consistency

Metaphor consistency, often also described as *causality*, denotes the quality of preserving the same spatial relations and underlying notions between the virtual world and the visualization metaphor. It provides a *continuity of experience* in a believable reality, which allows for comprehension, interaction and orientation. This hypothesis does not necessarily imply that the natural laws of time and physics have to be followed, but that the arbitrary rules implied by the virtual environment need to be continuous and explainable. In practice, this means that the data mapping mechanisms must be robust enough to visualize all possible data values, even when these are unexpectedly strange in any respect.

In practice, all representations that need to be believable for users must be continuously consistent. This notion poses important restrictions on the information mapping rules, yet simultaneously suggests the intrinsic ability to attract the user's attention to those parts of the representation that behave *differently*. Typically, these events will denote data patterns that differ from normal expected behavior.

In theory, the data filtering development phase removes or transforms those data entries that might break the infoticle representation consistency, taking into account the validity of the whole dataset. However, the filtering algorithms focus especially on maintaining the robustness of the visualization engine, rather than on averaging out unexpected data entries. In contrast, the infoticle method extensively employs metaphor consistency breaking to alert users for unanticipated data updates. For instance, the application prototypes showed that valuable dataset mistakes can be represented by infoticles leaving the scene, or by clusters that continuously behave in similar ways. Sudden trajectory alterations can be considered as more subtle consistency breaching patterns, as they represent the rhythmic causalities of the time-varying dataset. Static or steadily moving data objects have time-consistent data values, even when their pattern evolves over time, such as the star, comet and electron evolutions. In contrast, unexpected, and therefore potentially interesting data updates are mostly characterized by inconsistent patterns.

5.1.7. Causality

The infoticle simulation is driven by a continuous data update process. The emerging visual patterns hereby primarily result from data changes, and not from exact data values. In effect, the most challenging task is to link a change in spatial behavior to its factual cause, the update process behind it, so that pattern *variation causality* can be comprehended. One of the most unique capabilities of the infoticle system is its power to effectively represent various data update characteristics.

- **Galaxy.** Table 5.5 shows how possible data update frequencies and relative data value alterations result in distinctly recognizable patterns. It demonstrates how the infoticle method is best suited to visualize *characteristics of change,* instead of exact data values.

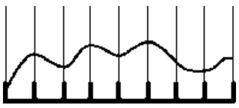
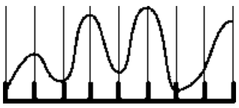
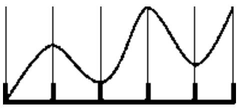
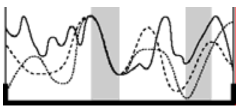
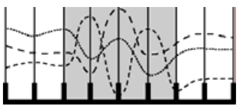
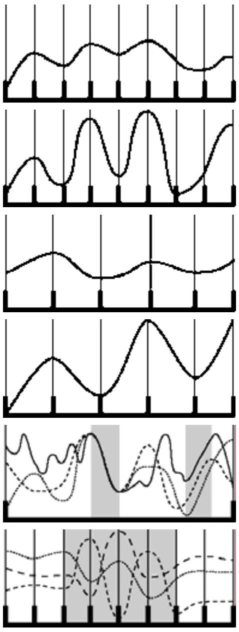
	Data Update Typology		Visualization
	Update Frequency	Relative Data Value Change	Pattern
	high	low	*comet*
	high	high	*quark*
	low	low	*electron*
	low	high	*star*
	any	equal	*time*
	equal	any	*burst*

Table 5.5. Visual Galaxy data update typology.

– **Boid.** Because only boids with equal relative data value alterations tend to group and the flocking rules maintain a constant clustering tendency, update frequency plays a less important role. Sequences of small data value changes offer equal boid infoticles more time to cluster, whereas slow and large data value changes result in boids escaping from the central flock. Infoticles that are subjected to rapid successions of large data value changes thus continuously attempt to cluster with different sets of mates, and often tend to stay steady in their current relative position, because the short and drastic spatial changes do not allow for enough time to cluster stably. Table 5.6 shows in detail the exact boid behaviors that result from different combinations of dynamic data update characteristics.



Data Update Typology		Visualization
Frequency	Relative Change	Pattern
high	low	*cluster*
high	high	*steady*
low	low	*cluster*
low	high	*expulsion*
any	equal	*cluster*
equal	any	-

Table 5.6. Visual Boid data update typology.

5.1.8. Interaction

Because of the complete novelty of the infoticle metaphor, one might easily draw false conclusions from the perceivable visual cues. Therefore, the interface design of the infoticle world has been kept as simple as possible, and shows the following features.

– **Cursor Focus.** The flashlight metaphor received many positive reactions and is considered as easy to use. By only showing detailed text labels in a small part of the display, the rendering performance can be easily controlled. Simultaneously, this concept also proved to successfully decrease the required human attention, as users can focus to a well-defined and limited region instead of being overwhelmed by text labels in their periphery view.

– **Continuous Overview.** Probably the most powerful feature of the infoticle method is the continuous contextual overview over the whole dataset. This feature prevents users from drawing possibly false negative conclusions, as humans tend to assume that only objects that are visible exist, a phenomenon also known as the *closed-world assumption*.

- **Contextual Zooming.** Because the infoticle world is three-dimensional and freely navigable, plus usable in an immersive virtual reality environment, users are able to maintain a natural feeling of orientation within the dataset. Contextual zooming in from global patterns generated by multiple infoticles to indiviual infoticle behaviors is accomplished by spatially zooming inside the virtual world. Because the immersive and stereoscopic human-size displays offer a huge perceivable space, the relation between individual patterns and the global context is never visually lost, even when the user zooms in on highly detailed regions.
- **Spatial Orientation.** Because the virtual world is inherently dynamic, users may get easily lost in space as they cannot rely on fixed navigational cues. However, both the restricted spaceball navigation paradigm and the static interface elements within the world help users orientate within the representation. As a result, only a few users expressed concerns about difficulties in way-finding or disorientation.
- **Meaningful Elements.** All elements within the world are meaningful and are directly related to the data representation. Decoration has been kept to a strict minimum. Moreover, all objects within the scene can be manipulated by users, so that their functionality becomes clear through a process of exploration rather than explanation. This idea of consistent simplicity has been chosen to avoid confusing users with elements that are solely motivated by aesthetic or real-world familiarity reasons.

5.1.9. Equilibrium

The infoticle visualization method differs fundamentally from traditional information mapping techniques, which translate data values directly and singularly into static spatial positions. Instead, a dynamic and evolutinary process produces a visual representation that is subjected to a constant state of flux. Distances between infoticles with equal directions or behaviors and between infoticles and forces denote data similarity. Spatial direction or behavior inequality means that infoticles are moving to other points in space, or are subject to different data update histories. However, these proximity measures need to be considered in relation to the phenomenon of *true representation equilibrium*.

Immediately after each data update, the complete representation is transformed into a state of extreme instability, as all infoticles with changed data values are influenced by new sets of tools and internal spatial behavior specifications. Consequently, a specific *equilibrium adaptation time* is required to reach a *true,* stable representation. A state of equilibrium is attained when all infoticles have reached their destination, and subsequently have acquired a constant dynamic behavior pattern. Such a true representation can be analyzed in a static state without the risk of misinterpretation. In practice however, the application timeframe is mostly shorter than the required equilibrium adaptation time, bringing the resulting representation under a continuous stress state.

The equilibrium adaptation time solely depends on the external environmental influences within the scene in case of the Galaxy World, and on the internal interdependencies for the Boid World. In practice, the required equilibrium time span $t_{equilibrium}$ is directly related to a single infoticle i at the greatest distance d_i from its attracting tool or from its mates with similar data values, which is traveling with velocity v_i to its final (tool or mates) destination T_i.

$$t_{equilibrium}(i) = \frac{d_{T_i}}{v_i} \rightarrow t_{equilibrium} < t_{application}$$

The equilibrium adaptation time can be employed to visualize *similarity in data history evolutions*. In theory, infoticles that represent similar data values over time, and are consequently already close together, receive relatively more equilibrium adaptation time to cluster than those that were subjected to a considerable amount of different data values within this timeframe, and are further away as a result of this. Therefore, the continuous dynamism of the data-driven particle methodology automatically clusters those infoticles that have equal *data update histories*. This is especially true for the Boid scenario, as the cluster tendency in the Galaxy World is driven by contextually independent behavior rules. In effect, infoticles with equal data developments over longer periods of time are clustered by exploiting the adaptation time span needed by the visualization system to reach a stable representation.

5.2. Evaluation

This section evaluates the infoticle metaphor in comparison with the most common alternative time-varying information visualization methods. Furthermore, its data-mapping paradigms and interface are assessed against various design guidelines listed by other researchers. This *comparative analysis* provides an overview of some of the qualities and limitations of the infoticle metaphor, especially in relation to other similar visualization approaches or, more generally, to relevant user interface and user application evaluation models.

5.2.1. Methodology

Many potential users attempt to compare the infoticle method with other, more common information visualization approaches. However, it is not easy to find real-world examples that can be directly compared with data-driven particles, also because the field of information visualization does not particularly focus on dynamic visualizations or time-varying datasets. The following list categorizes and compares some of the currently available alternative methods for time-dependent data visualization.

– **Static State Replacement.** Most information visualization approaches represent data value updates by instantly replacing a static world with another. Consequently, users are unable to compare quantitative evolutions or follow evolving tendencies. This method is especially impractical for dynamic datasets with slow update frequencies or large data value changes, because the continuous sequences are perceived as discrete steps. This approach is typically adopted because the static world generation requires considerable calculation time and interpolation or morphing algorithms are relatively difficult to implement.

– **Static State Morphing.** Time and information visualization can be merged by filtering what is shown by the selected time period and updating the display as the selected time period changes, a method also called dynamic queries (Shneiderman, 1994). Users can then fly-through these information spaces by incrementally adjusting a query with sliders, buttons, and other filters while continuously observing the changing results. Although such techniques are often limited to normal visual cues, motion can be included as well. These approaches animate the data visualization between different static states that accurately represent the data values within discrete time intervals. Mathematical interpolation algorithms simulate the data elements between their start and end positions and convey the feeling of *progress*. Morphing methods attempt to maintain the overall context and orientation while demonstrating the data evolution between known points in time. By animating instead of suddenly repositioning elements, users can keep track of objects and perceive the differences in time. A global interpretation of data tendencies is often made impossible due to the multitude of resulting motion typologies. These techniques differ from data-driven particles because they require pre-computations of the static states, and thus are unable to visualize real-time data. Also, motion typologies are not regarded as informational, so that it is the object moving, and not the nature of the motion that denotes the change in time.

– **Equilibrium Attainment.** Force-directed diagrams and self-organizing maps show many conceptual similarities with infoticle systems, as they are internally controlled by local interactions and only reach a state of equilibrium after a certain adaptation time. The majority of force-determined methods focus on the visualization of static datasets, and need pre-computed data similarity matrices to determine the spring strengths between pairs of points. Such matrices store pairwise the relative mathematical interdependence between all the data objects within a representation. Next, a randomized start constellation is chosen, after which an equilibrium state between all points is calculated iteratively. Notably, force-directed visualizations only represent informational values by individual distance differences, and do not generate sets of recognizable dynamic behaviors. Therefore, movement characteristics have no specific meaning, and because of the randomized start conditions, user-directed time simulations are difficult to implement and specific representations can often not be recreated. Force-directed representations require either dedicated pre-computation or a reasonably long equilibrium adaptation time span to reach a final, true representation. Unlike in the infoticle method, this timeframe is not used to cluster similar data histories.

– **Time-Series Plots.** More traditional approaches use time-series plotting, connecting sets of static states that are mapped in space and time with simple curves, stacks or timelines, such as stock market chart line diagrams and web usage bar charts. Although these images usually illustrate changes accurately and understandably, they are unable to effectively represent large quantities of data objects and attributes simultaneously, as many graphs will overlap. Moreover, all data objects need to be converted to an equal spatial scale and data dimensionality, limiting such time-series plots to specific data object types and data value thresholds. Technically, the most restrictive factor of such methods is the limitation of available display space and resolution to effectively represent all occurring data tendencies and dimensions in time within a single image.

– **Threshold Activation.** Triggering mechanisms alert users when unexpected events take place between specific, predefined thresholds. Such approaches can be found within many fields, such as financial broker software tracking the real-time performance of company quotes or network administration applications that detect hacker access patterns. Although these alerts could effectively replace typical infoticle patterns that denote exceptional data performances such as quarks, comets or expulsed boid infoticles, they are representations for already known and predicted data patterns, are unable to present long-term tendencies, give a continuous overview over the whole dataset or trace back data trends in history at different contextual levels.

Conclusively, the infoticle method offers many features that are truly unique, transcending most alternative approaches. Instead of pre-calculating and animating static representations, data updates are categorized by instantaneously interpreting data-dependent local interaction rules within the representation space itself. Furthermore, users are able to have a continuous overview over the whole dataset, manipulate the representation and animate the time simulation.

5.2.2. Visualization

When an interactive system is well designed, the interface almost disappears, enabling users to concentrate on their work, exploration or pleasure. In fact, users hardly commented the interaction paradigms or interface design, but rather seemed to require support in regarding understanding the dynamic infoticle characteristics with regard to data features. A difference was detected in understanding the informational values of visual patterns between users with knowledge about the dataset and so-called non-experts. However, the visual patterns built by the infoticles and their timeline ribbons were easily perceived by all users as different data update typologies, proving that data-driven particles can visually sort data by dynamic spatial behavior.

The following list of desirable representation and interface properties is inspired by different approaches that can be found in literature (Erickson, 1990, Young and Munro, 1998, Sprenger, 2002), and highlights some of the qualities and limitations of the infoticle visualization method with regard to the system features.

- **Scalability.** Adaptability to large amounts of data of high dimensionality. The application prototypes proved that the infoticle method is capable to visualize large amounts of data with many data entries spanning over long periods of time.
- **Flexibility.** Handling different types of input data without configuration effort. The infoticle methodology is applicable to many different data typologies, and is even capable to merge multiple data types simultaneously within a single visualization, although some practical implementation restrictions do apply. Because of the current experimentation stage, most necessary reconfigurations are situated on the software code level, as no user-configurable framework has been implemented yet.
- **Extendibility.** Independence of data-specific interaction and visualization methods. The described prototypes all use equal interaction techniques while utilizing the same core features of data-driven particles. To enhance data pattern recognition, small adaptations are required depending on dataset characteristics. Simultaneously, the infoticle method is still extendable with novel ways of mapping dynamic data updates onto specific infoticle attributes or spatial behavior patterns. In fact, many users almost naturally start inventing new ways of visualizing data while using the infoticle metaphor.
- **State-of-the-Art Technique.** Appealing to the cognitive system while taking advantage of contemporary graphics hardware performance. The infoticle system clearly exploits currently available calculation as well as graphics power. The performance bottleneck consists of updating data values and coordinates of thousands of infoticles rather than graphically rendering the resulting three-dimensional movements. Several algorithmic optimizations could be implemented to tackle this issue, although the outcome will not necessarily be noticeable. Simultaneously, the infoticle method uses the latest presentation technologies offered by immersive virtual reality technology. In effect, stereoscopic and immersive perception has been exploited to present users with large amounts of moving points in an effective and understandable way.

- **Clustering & Hierarchies.** A multi-resolution setup to break down the complexity of data, such as clustering and interactive level-of-detail. The infoticle method provides a global overview over the evolving dataset, while enabling users to focus on behaviors of individual data objects. The continuous presence within the same three-dimensional world guarantees a contextual overview. Furthermore, the complexity of time-varying datasets has been overcome by simulating data evolutions gradually in time.
- **Portability.** Independence of operating systems or established standards. Currently, the infoticle method has been implemented using the OpenGL Performer and blue-c API. Although this software is available for most operating systems and hardware configurations, the infoticle methodology itself still requires the combination of quite powerful computing and graphics hardware. However, the findings of this research form an ideal platform to test the viability of the visualization method on normal desktop configurations and, for instance, in a purely two-dimensional environment.
- **Amount of Structure.** The infoticle metaphor brings forward a large amount of concepts and ideas that can be used to help explain the data model that the representation provides. In spite of its obvious simplicity, the descriptions in this thesis demonstrate the amount of aspects that are associated with data-driven particles and their emergent visualization patterns. Furthermore, the cognitive qualities of motion offer users a rich palette of interpretation possibilities related to data update typologies.
- **Applicability of Structure.** The infoticle metaphor is relevant to the problem of time-varying data visualization, because many dynamic datasets consist of fast-changing data entities that show some relationship to each other, be it in terms of data value, data value change, data value history or data update frequency. However, the dynamic traveling of infoticles within a virtual world might misleadingly suggest to some users that data entities are physically transported between specific spatial locations.
- **Representability.** The infoticle metaphor employs some of the latest animation and behavioral algorithms, and has been especially created to be displayed within stereoscopic environments. However, the use of other media can be considered for smaller datasets or two-dimensional worlds. Currently, the infoticle method focuses on visual artifacts, although other non-visual means, such as auditory cues that help users to discover infoticle data clusters or movement directions, can be imagined as well.
- **Suitability of Audience.** Unfortunately, some users who are unfamiliar with the dataset have difficulties understanding the emerging patterns. Nevertheless, typically little explanation is required to help users comprehend the world-governing rules, and persuade them to explore the dataset.

- **Extensibility.** Many aspects of the data-driven particle methodology can be easily extended. Both the behavior rules and interaction mechanisms can be broadened and fine-tuned depending on specific dataset characteristics. Even alternative particle attributes, such as *mass* and *friction*, can be merged with the rule-based visualization logic. Because of the unpredictability of the behavior rules, some data patterns can be detected that were not expected to occur.
- **Individuality.** Fundamentally different system components, such as the interface and the data-representing artifacts objects, appear in a different dimensionality, and can thus be easily distinguished.
- **Distinctive Appearance.** Differing data patterns appear in a contrasting and recognizable way, as a result of distinct motion typologies and shape generations.
- **Automation.** Data updates are automated, although scenario development and parameter fine-tuning is still accomplished on the source-code level.
- **Resilience of Change.** As a very important feature of time-varying information visualization, small data changes result in small visual alterations and vice versa.
- **Visual Complexity.** Although the dynamic nature might render the resulting scene visually complex, the understandable, static layout and the intuitive navigation lower the strain for users to comprehend and interactively explore the meaning of the infoticle representation.

5.2.3. Dimensionality

It should be noted that each infoticle is limited to be externally influenced by a single force within a particular application timeframe. Like mentioned before, the infoticle methodology solved the problem of representing simultaneous multiple data values in a single database timeframe within parallel sequential datasets by calculating a so-called *average force* (see Section 3.5.2. Infoticle – Simulation – Update). In fact, other solutions for this *data dimension parallelism* phenomenon can be imagined, although experience has shown that these increase the visual complexity and thus the understandability of the resulting emergent visual patterns.

- **Statically.** Multiple dimensions can be merged either by creating a dimensionally weighed average force that is influenced by different sets of forces, each representing an extra data dimension, or by introducing several individual forces with attraction strengths that are dependent on the data dimensions. In effect, such an approach transforms the three-dimensional space into a coincident multi-dimensional world. However, users need to compare the resulting multi-dimensional average force to the relative positions of the different force sets without knowing which forces have been included in the calculation and with which relative attraction strengths. Like Figure 5.5 demonstrates, it is practically impossible to meaningfully and faultlessly comprehend the resulting average force position.

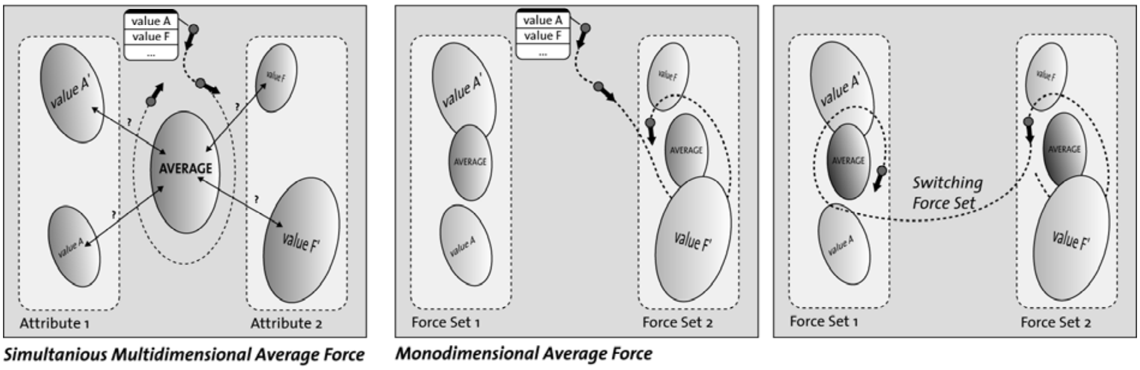


Figure 5.5. Multi-dimensional average force.

– **Dynamically.** Infoticles could be attracted by more than one force simultaneously, even for mono-dimensional datasets. However, the implementation of multiple, concurrently attracting points often results in highly complex, unpredictable and even chaotic behaviors that have the potential to propel the infoticles out of the influence ranges. In fact, the introduction of additional forces dramatically alters the curvature and thus the understandability of the emergent trajectories. In practice, the accurate, stable trajectory prediction of points attracted by multiple masses following gravitational principles is a well-known mathematical problem, also known as *N-body dynamics*. The amount of possible parameters within three-dimensional space leads to a *non-integrable* equation. Consequently, such trajectories are rarely stable, let alone circular or elliptical in nature. In addition, Figure 5.6 shows what would happen even if this problem were solved, for instance, thanks to a strict rule behavior system that successfully regulates the trajectory stability. The resulting orbits encompassing multiple forces show the intrinsic danger of misinterpretation when nearby but unrelated forces are unexpectedly enclosed as well.

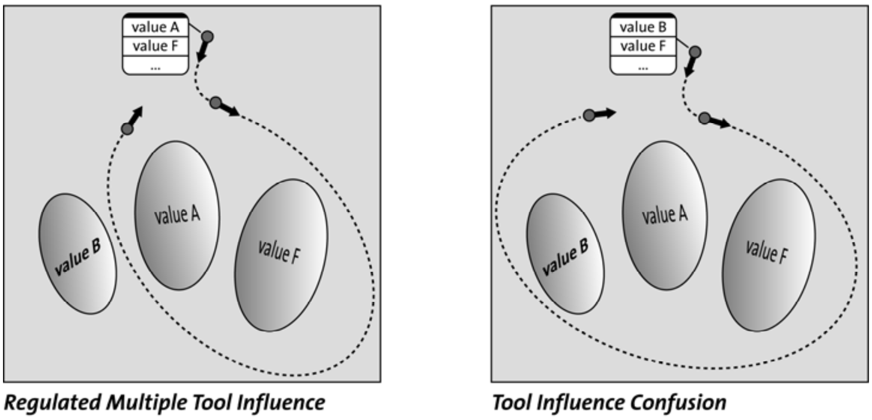


Figure 5.6. Multiple force attraction.

Users tend to interpret movements in relation to *singular causalities* only, and do not intuitively relate to the interrelationships of more than one data dimension, or to more complex physical attributes such as object mass or center of mass. Although this perception phenomenon might be solved by implementing extra behavior rules that stabilize unstable trajectories, it still remains difficult to comprehend the precise amount and exact positions of multiple influences out of simple dynamic behaviors or spatial orbits. Therefore, it was deliberately chosen to decrease the required cognitive effort by limiting the degree of employed visual complexity. For this reason, the current infoticle method restricts continuous simulation sequences to mono-dimensional data.

5.3. Usability

The term *usability* addresses the intended use of a tool to accomplish tasks in the best way possible, so that users are able to employ it effectively. Some typical ways to improve usability include: shortening the time to accomplish tasks, reducing the number of mistakes made, reducing learning time, and improving people's satisfaction with a system. The following section describes the usability issues that relate to the infoticle methodology, considered as a tool to effectively visualize time-varying datasets. Consequently, it explains in what contextual circumstances the infoticle metaphor is ideally used and produces the most successful results.

5.3.1. Dataset

The application prototypes presented in Chapter 4 have demonstrated the infoticle method's flexibility to represent a multitude of datasets. However, the data-driven particle metaphor is not capable to represent *all* possible time-varying datasets, as it handles specific dynamic data characteristics more efficiently than others. Experience has shown that an ideal infoticle dataset includes the following features.

– **Size.** The dataset should contain many data objects so that a meaningful and dense visual representation with comparable patterns can be generated. Small amounts of data objects produce relatively empty worlds with large-scale patterns that do not denote any global tendencies and can be represented more effectively by other methods. A typical infoticle world consists of thousands of unique data objects that reoccur often within the database timeline and are subjected to successive data value alterations. Performance issues generally occur when the visualization system is unable either to efficiently look up data values within an extremely large data cache or to update all infoticle movements effectively because of an extremely large infoticle collection. In contrast, a boid world dataset generally counts at least ten times less individual data objects due to the exponential calculation-intensive evaluation of all neighboring states for each single infoticle at each frame. In this context, mentioning precise numbers is not relevant, as these are solely related to computing hardware capabilities. Ideally, an infoticle dataset thus contains a *relatively large*, but not massive, amount of *regularly reoccurring* data objects.

– **Duration.** The dataset can be of *any* duration and is only practically limited to the time span that users are expected to explore the dataset. Long dataset timelines have to be artificially shortened by larger database timeframes or smaller application timeframes, with the inevitable risk of missing smaller scale data evolutions. Generally, users explore a dynamic simulation for less than 15 minutes, hereby practically reducing the effective timeline duration that can be shown. In general, the dataset should contain at least hundred times more data entries than the desired database timeframe duration, so that in any case a sequence of hundred time updates can be shown.

– **Attribute.** An ideal infoticle dataset consists of *numerical* data values, as these are more efficient to handle and represent than nominal data entities. Numerical data attributes can be easier compared and mathematically averaged, although no real performance gains have been determined yet. In fact, the infoticle method currently handles nominal data values as effectively as numerical data values by labeling reoccurring text values with unique numerical identifiers.

- **Dynamism.** A considerable amount of data objects should be updated within each database timeframe, as otherwise the dynamic representation would contain a lot of irrelevant, static data objects that disturb an effective pattern perception. The amount or frequency of updated data objects does not have to be constant, as the infoticle method is suited to handle differences in update frequencies and data object quantities. In practice, this means that a dataset should contain a *substantial but changing* amount of both static and time-varying reoccurring data objects at each data update.
- **Timeframe.** Extremely frequent data value alterations within a single database timeframe increase the data processing and communication efforts dramatically. Therefore, the update frequency or amount of updated data objects ideally should *not* drastically *peak* at any point in time. As mentioned before, performance issues are highly dependent on the used hardware configuration.
- **Dimensionality.** Ideal infoticle datasets contain singular or few simultaneous data attribute dimensionalities. As users have great difficulty to comprehend resulting dynamic patterns generated by multi-dimensional dependencies, the infoticle method limits the visualization to a single data dimension at all times, and visualizes data-dimensional relationships by means of singular *data dimension switching*.
- **Flow.** In spite of the data dimensionality restriction, the infoticle method has the potential to visualize other time-dependent dimensions that are often overlooked or even unavailable in most time-varying datasets, namely the detailed *flow* (or *successive or time-based change* in data attribute dimensionality) of data objects in time. Like the Electron World prototype demonstrates, infoticles are capable to effectively visualize the continuous flow between data dimensions in an accurate and intuitive way.
- **Value Condition.** For reasons of perception effectiveness, it is often chosen to limit the amount of available forces within an infoticle scene. As a result, an infoticle visualization is capable to represent a relative small amount of data value conditions. When more data conditions are required, all numerical and nominal values are grouped into specific categories, each corresponding to a single influencing tool. This process of *data condition grouping* lowers the level-of-detail, but increases the amount of occurring data frequencies, and thus of distinguishable infoticle patterns.
- **Noise.** The infoticle method is capable to handle non-precise datasets effectively, as noisy data entries are *flattened out* either by average force calculations or local interactions within the infoticle collection itself. In effect, if most data objects are subjected to equal value disturbances, the emerging visual patterns are not significantly affected. Especially the boid method has the intrinsic quality to remain stable, as its flock centering force of data-similar infoticles stays relatively unchanged during small data value alterations (see Section 5.1.7. Analysis – Grammar – Causality).
- **Real-Time.** Typical infoticle datasets contain time-stamped entries of specific reoccurring data values with constant data attributes. Whether this dataset is collected beforehand and thus represents the tendencies of an *historical* timeline or is continuously *streamed* and stored in the database in real time by some external update process, does not play a substantial role in the context of an infoticle visualization.
- **Type.** A dataset can include any possible data type, as infoticles are not dependent on the data entry characteristics and can thus represent many data types simultaneously. However, multiple datasets that are combined within a single infoticle data representation ideally should have at least *one data attribute in common*.

Conclusively, an ideal infoticle dataset consists of a reasonably large quantity of data objects that are updated frequently during a relatively short database timeframe, containing either few data dimensions or a precise description of the time-varying change in data dimensionality (flow) over time, a limited amount of possible numerical or nominal data values, and may include some degree of data inaccuracy (noise). Galaxy scenarios are especially suited to explore the time-varying evolution of nominal datasets, whereas infoticle boids are ideal for representing rapidly changing numerical datasets.

5.3.2. Medium

The three-dimensional infoticle visualization creates spatially complex constellations involving large amounts of moving points, which are difficult to interpret at first glance. However, the perceptual aids offered by a virtual reality environment are extremely helpful when inspecting these structures. As mentioned before (see Section 3.2. Infoticle – Cognition), the three-dimensional world increases the amount of points that can be effectively perceived without occlusions and allows for a smooth and continuous detailed view from arbitrary viewpoints without loosing a contextual overview. The stereoscopic feature enables users to identify three times as many data points as on normal displays, whereas immersion enhances the perception of motion and the subsequent interpretation of the resulting trajectories. The human scale interface elements remain static within the dynamic world, offering users fixed cues of spatial orientation. All elements within the world play a meaningful role and interact with the representation to build up a believable abstract reality governed by consistent metaphor rules. As a result, users are literally surrounded by a world that is determined by dynamic data processing and is purely built up out of data.

However, one can imagine the core infoticle method to be used on other, less sophisticated presentation media. In this case, the following aspects should be subject to a thorough reevaluation.

– **Interface.** Applications used in non-immersive environments can employ menus and widgets as users do not need to retain a consistent spatial orientation and can operate more user-friendly input devices. This adaptation could transform the infoticle method to a fully configurable visualization authoring tool for many meaningful purposes.

– **Configuration.** Because of less restricting interface possibilities, infoticle parameters can be altered at runtime, enabling a more detailed and direct experimentation with e.g. metaphor principles or tool data value conditions.

– **Dimensionality.** At least in theory, the infoticle metaphor could be used in flat, two-dimensional worlds instead of in a three-dimensional believable virtual reality. This restriction would not only decrease the required perception efforts, but would also port the data-driven particle principles to less sophisticated hardware configurations. However, a non-stereoscopic data representation visualizes three times less data objects, purely relies on cue-of-motion to detect depth and behavioral differences and suffers from visual occlusions. A further reduction to a flat, two-dimensional canvas decreases the amount of points that can be effectively recognized, increases the risk of overlapping infoticles and limits the possible behavior typologies that can be spatially generated because of the lack of the depth dimension.

- **Functionality.** Because applications developed for traditional displays need not to solve many complex virtual reality interaction and interface issues, supplementary tasks such as data retrieval and database querying could be easily added to the visualization system, although corresponding visual analogies need to be invented for these actions.
- **Metaphor.** The reduced pattern recognition effort enables a more elaborate interpretation of the infoticle metaphor, allowing for infoticles to be attracted by multiple forces, forces to move in space, or boids to be controlled by individual goals.

Conclusively, the possible outcome of combining the infoticle metaphor with other presentation media can be predicted with some certainty, but not proven without implementation of several demo-oriented test cases.

5.3.3. Metaphor

Most of the previously mentioned application prototypes were inspired by the motivation to explore the validity and qualities of the infoticle visualization method. However, it is not coincidental that an increasing abstraction tendency can be detected in the successive visual outcomes. Through a process of exploration, the metaphor was gradually abstracted to its core features, so that interaction possibilities were excluded in favor of additional local interaction and more subtle pattern generation. Subsequently, these findings would form the ideal foundation to implement a simplified visualization method from *bottom-up*, instead of following a *top-down* development methodology as described in this thesis.

According to a common definition of information visualization, novel visualization metaphors should lessen the complexity of the data that is perceived. Theoretically, the use of three dimensions adds an element of familiarity and realism to visualization systems. When data representations resemble the real world, users require less cognitive strain to comprehend the informational meaning. Possibly, the data-driven particle method might not directly obey this rule when users without prior dataset knowledge evaluate the emerging visual cues. Because the data mapping mechanism does not translate data values onto directly perceivable visual cues, some cognitive effort is needed to detect the connotations behind the occurring events. In effect, this information visualization metaphor does not directly involve a real-world analogy that is based on everyday knowledge or is understandable at first glance. On the contrary, this method appeals to a more subconscious and *aesthetic interpretation* of motion, causality, form recognition and spatial clustering.

To a large extent, the human-computer interface of an information system determines its usability. Other factors include the performance of the computer hardware on which the system is used, general ergonomics during interaction, or perhaps even the contextual environment. According to Nielsen (1993), a renowned interface usability expert, usability is associated with the next five attributes.

- **Learnability.** It has been recognized that the infoticle metaphor requires a certain amount of explanation and hands-on experience to be used effectively. Especially the input device is not simple to use, and has the tendency to be tiring as the users have to continuously hold it up and point without any mechanical support. The system itself is easy to learn so that users rapidly can get started to accomplish some useful work.
- **Efficiency.** Once having run through the first phase, users are able to use the infoticle visualization in a productive way. The navigation paradigms enable manipulation and observation from all angles and distances. However, the number of potential users is relatively small and specialized, and the targeted application field is reasonably narrow, so that the time spent with the system is mostly short and sporadic.
- **Memorability.** Once the core idea of data-driven particles is understood, users can easily remember it and return to the system after some period of non-use. However, the odd input device mostly requires some adaptation time before every re-use.
- **Error.** Users tend to interpret the emergent visual pattern correctly. Navigation and interaction rarely produce mistakes and users can easily recover from them. Because the data feed generated by the timeline simulation cannot be turned back during the visualization, all tool interactions have direct repercussions on the evolution of the representation, even when the simulation is animated backwards.
- **Satisfaction.** Most users regarded the system pleasant to use, and are subjectively satisfied when utilizing it. Most negative comments relate to the effort of getting used to and employing immersive virtual reality technology for the task of visualizing abstract datasets.

5.3.4. User

Some early user testing has been performed with colleagues, the people involved in the project or those present at several presentations on public, corporate as well as academic levels.

Users with backgrounds in abstract realms generally were more enthusiastic than those used to handle problems in a physical real-world context. The application seemed to cause a high degree of user engagement, as users liked to use it over longer periods of time and appreciated the overall interface design. Most users did not encounter any form of disorientation, a very common problem in three-dimensional virtual worlds in general and immersive worlds in particular. The trackball navigation paradigm requires some adaptation time as users, for instance, cannot reach the backside of the data representation, but rather have to turn to whole world by 180 degrees.

In contrast, some users seemed to be overwhelmed by the continuous dynamics of the infoticle representation. Because of the many contextual factors involved in an immersive virtual reality installation (e.g. unknown input devices, stereoscopic vision, huge display surfaces, spatial sound effects, etc.), the exact cause of the phenomenon is still unknown. No practical possibility exist to compare this behavior directly with other applications on the same hardware setup, as most other virtual reality applications either pursue a relatively passive experience, have entertainment purposes, or do not represent time-varying data. In fact, in general, few virtual reality applications exist in which users have to perform real-world productive tasks.

At closer examination, the infoticle metaphor core might encounter some usability issues that cannot be easily fixed.

- **Reversibility.** Because the system requires a continuous adaptation process after each data update, the reversibility of user actions is not always guaranteed. This condition, one of the direct manipulation guidelines (Shneiderman, 1998), has only been applied in the Modeling and Electron World, in which users are capable to replay indefinitely the same time loop and adapt the tool constellations until they are satisfied. In the other application scenarios however, manipulating the tools also alters the representation outcome due to the gradual external influence on the infoticle collection.
- **Configuration.** Although information exploration by definition starts without any mental model about the dataset, the infoticle method still requires some prior user expectations in order to define the exact tool data value conditions before system initialization. Unfortunately, these tool conditions cannot be altered at runtime for reasons of inefficient text interaction within virtual reality environments.
- **Quantity.** Users are neither physically nor cognitively able to simultaneously follow large amounts of infoticles, especially when they spatially drift too far away from each other. However, users are able to select a handful of infoticles by brushing and changing their color, so that they become more easily perceivable. Also, multiple infoticles can be followed in clusters or events can be traced by replaying timeframes at different speeds.

Except for sporadic, small evaluations, no real user-related usability testing has been performed, partly because of the small size of the potential user community and the number of people with specialized dataset knowledge, partly because the infoticle metaphor has been in a state of experimentation and method exploration for a relative long time.

5.3.5. Usage

The infoticle method is especially suited to visualize datasets little is known about initially, and for which some sort of data model still has to be developed. It is one of the few approaches that are able to effectively represent time-varying tendencies on a global and individual scale simultaneously. However, the visualization metaphor also shows some usage issues potential users should be aware of beforehand.

- **Value.** As mentioned before, data values determine spatial location whereas data updates generate dynamic behavior. The infoticle method is not particularly suited to represent *individual* data values effectively without comparison to neighboring infoticles or proximate tool data values. However, exact data values become readable when the corresponding infoticles are highlighted or selected.
- **Outlier.** The infoticle method is particularly suited to filter out specific data value exceptions in the context of all other simultaneous data value entries within the dataset, also called *odd-performers* or *outliers*. Especially boid infoticles behave in a stable way when the majority of data objects undergo an analogous data evolution, whereas the Galaxy metaphor is rather based upon the visual grouping of individually determined but still equal motion typologies. In contrast, odd-behaving atomic entities are ejected away from clusters or show a dynamic behavior that is fundamentally different from the rest. By doing so, the infoticle method is literally capable to visually *filter* out the great majority of *non-interesting data values* within a specific timeframe.

- **Exception.** The infoticle method is particularly suited to represent *oddities* in individual and global *data update frequencies*. Such dynamic update characteristics become typically detectable by the continuous but sudden changes in infoticle velocity directions or by infoticle trace irregularities.
- **Trend.** The effective perception of time-related trends within the data considerably depends on the *granularity* of the chosen *application and database timeframes* in relation to the database timeline. Whereas some timeframe configurations are ideal for highly detailed investigations of short time units, other configurations might not show any significant dynamic pattern within the same dataset.
- **Interpretation.** A few resulting infoticle patterns are *not intuitively interpretable*, particularly those appearing in the boid infoticle scenario. Although spatial phenomena denote significant global changes within the dataset, it remains the responsibility of the observer to explore the data with other means to find out the underlying driving forces.
- **Attention.** Infoticles rely considerably on the attention gathering feature of motion. Important events that might get lost in alternative visualization contexts become immediately noticeable by popping-out effects that catch the eye of the observer. However, this feature shows the inherent danger to overwhelm users visually by the continuous presentation of dynamic elements. *Motion attention grasping* is particularly useful when visualizing real-time data, as the user is immediately visually notified when significant data tendencies occur.
- **Non-Conclusive.** The infoticle technique is basically *non-quantitative* and *non-conclusive*, as no immediate numerical measurements can be made for the majority of the emergent patterns. This means that visualization goals related to rankings or immediate comparisons do not match well with the infoticle system. Instead, infoticles appeal to the less graspable intuition of users to link visual patterns with data interpretations. In effect, this feature was particularly appreciated by users, as the major goal of the prototypes consisted in the detection, demonstration and subsequent analysis of fundamentally unexpected patterns rather than producing mathematical conclusions that generally are considered irrefutable pieces of evidence.
- **Flow.** The infoticle metaphor is especially appropriate for visualizing the sequential transformation of data from one dimension to another over time, demonstrating valuable tendencies that are often overlooked by most data-gathering mechanisms. In practice, this means that infoticles are suited for datasets that represent some kind of traceable *flow between data attributes*, which can be represented effortlessly by spatially streaming infoticles from and to force sets each depicting a single data dimension.
- **Mining.** Infoticles are rarely suited to perform precise data mining operations, or to find atomic data entries within the dataset at execution-time. As the filtering granularity of the filters must be relatively high to produce a perceivable effect on the infoticle collection, *atomic filtering* is rendered difficult. However, predefined data objects can be perceived more easily by coloring the corresponding infoticles in a contrasting color.

- **Comparison.** Individual data object performances can be easily *tracked* by *selecting* and subsequently *coloring* corresponding infoticles. Such infoticles are immediately spotted, even in large collections, and can be dynamically traced. This brushing feature is especially useful when users are interested in a handful of data objects, but deteriorates when users select more infoticles than the amount of effectively distinguishable moving points in three-dimensional space. However, even large amounts of such infoticles can be easily retraced and compared when halting and replaying the simulation.
- **Type.** Infoticle systems are able to combine fundamentally *different data types* within the same representation world. Because no singular spatial data mapping mechanism is involved, infoticles representing different datasets do not need to relate to Cartesian scales, space dimensionality or specific visualization features. Instead, data types stream out of different infoticle sources and literally share the same infoticle space, tools and behavior rules.

5.4. Conclusion

The infoticle method is viable to visualize most dynamic dataset characteristics. The analyses within this section have shown the different limitations and strengths of this methodology, all which can be summarized as follows.

- **Pro.** Many of the infoticle advantages are novel and analogies cannot be found in other visualization approaches.
  - **Metaphor.** The infoticle method is capable of visualizing large, time-varying datasets containing both numerical and nominal data entries, which are subject to changing update frequencies that may alter the varying quantities of data values in any way. The visualization continuously shows both individual and global tendencies by a process of navigating and spatial zooming that retains the contextual overview and enables a true mental and virtual immersion within the data. The resulting data patterns denote the *data update frequency*, *data value alterations*, *data value histories* and *data update histories* of individual data objects along with the corresponding similarities between multiple data objects. These patterns are not determined by predefined singular relationships between possible data values and visual cues, but rather combine interdependent visual cues that either emerge out of local interactions or are controlled by local and hierarchical interaction rules. Such rules control the spatial direction, speed, color and behavior depending on the represented data values.
  - **Use.** The infoticle visualization method is especially suited for exploring datasets of which little is known beforehand. Data patterns can be detected *dynamically* as well as *statically*, and can be influenced by a process of *direct interaction* through an interface merged with the visualization scene. Users are able to replay the timeline simulation at different speeds and in distinct time directions. The whole system is completely adapted for use in immersive, stereoscopic virtual reality environments. The metaphor is still open to adaptation or extension to other data mapping paradigms, and has shown a great potential to engage users.

- **Contra.** Most disadvantages of the infoticle metaphor relate to its extensive use of motion properties and sophisticated hardware devices.
  - **Metaphor.** To avoid visual complexity, the visualization method restricts the representation to a single data dimension at a time. Because the whole dataset needs to be replayed by a continuous simulation process, the duration of the dataset is limited to the expected application duration or desired data visualization granularity. The nature of data-driven particles makes the recognition of exact data values difficult, as the metaphor is specialized in showing *evolutionary tendencies* instead.
  - **Use.** Because of the combination of immersion, stereoscopic perception, virtual reality display technology, novel input devices and the continuous dynamism of the data visualization, some users have expressed an overpowering feeling. During the adaptation time needed by the system to reach a true representation after a data update, false pattern conclusions might be drawn. The continuous update process and particle animations require sophisticated calculation hardware with parallel processing power. The required hardware configuration and specialized large, time-varying datasets reduce both the amount of potential users and total utilization duration.

Furthermore, previous comparisons have shown that the data-driven particle method is able to represent time-varying datasets unlike other traditional approaches, demonstrating several unique visualization qualities. The *equilibrium adaptation time* required to reach a true representation might initially lead to potentially false interpretations, but also facilitates the clustering of time-dependent tendencies, such as similar long-term historical trends.

6. Discussion

There is a need for novel information visualization methods that enable the perception of patterns and tendencies within time-varying data. Although such dynamic datasets inherently contain various meaningful and valuable characteristics, their effective representation has not been extensively explored.

Information architecture typically deals with the effective organization of data in space, merging architectural principles and aesthetic considerations with the abstract world of information processing. Therefore, it is the ideal field to design enjoyable three-dimensional worlds that are constructed solely out of internal data structures, incorporating intuitive and enjoyable interfaces to facilitate information handling.

Because of the large quantity and complexity of data, new information representation modalities are required to offload some attentional effort to more efficient perceptual processing. While most contemporary approaches examine the communicative properties of traditional static graphs, the visual dimensionality of *motion* remains obscure. However, findings from cognitive science have shown that motion possesses some compelling features that make it potentially useful for visualizing information. *Motion typology* is a significant feature for recognizing *behavior* differences, as both the shape and the direction of a motion path facilitates the visual grouping of moving objects. In addition, empirical research has proven how motion cues are able to increase task engagement and enables a continuous contextual information display.

Immersive virtual reality technology becomes increasingly mature and user-friendly, presenting the unique qualities of spatial orientation and stereoscopic vision to a wide, non-expert audience. In effect, this sophisticated display technique introduces several cognitive features that enhance the effective perception of both space and motion, while offering application designers a tremendous range of possible representation and interaction methods. In fact, many researchers still anticipate meaningful virtual reality applications that shift away from purely scientific motivations and productivity reasoning to incorporate the concepts of *user engagement* and *explorative interaction*. Unfortunately, only a few novel information visualization approaches are being investigated within the virtual reality realm, probably due to the relatively complex implementation procedures and the multitude of related contextual parameters.

Although *particle systems* are a well-known technique for visualizing physical, dynamic and unstable flows, they have never been used for representing abstract, time-varying datasets. Instead of being driven by complex mathematical formulas that attempt to simulate real-world phenomena, data-driven particles follow simple local rules that react upon continuous data updates. External elements within the scene influence infoticles with matching data values, whereas internal interdependencies adapt infoticle attributes in parallel. As a result, different dynamic and static patterns *emerge* that can be cognitively interpreted and subsequently linked to corresponding structural data characteristics. The parallel execution of *self-organizing* principles facilitates a consistent representation of complex, large and time-varying datasets in a visually expressive way. The straightforwardness of this concept makes it applicable to various visualization goals in many different contexts.

6.1. Insight

This thesis demonstrates how the combination of virtual reality technology and cognitive principles is exploited to generate a dynamic information visualization that simulates the evolution of abstract data in time. It has described how the infoticle method combines the concepts of *spatial similarity*, *metaphor consistency* and *pattern causality* to build up a coherent and interpretable virtual space representing dynamic information. As a result, many issues mentioned in this thesis may be valuable for other research initiatives that deal with visualizing time-varying datasets in particular, or need to implement virtual reality applications that process abstract information.

- **Cognition.** One of the main themes underlying this thesis is the value of discovering and using design principles derived from *cognitive science*. This field does not only serve as the theoretical foundation of many metaphor design decisions, but also functions as a rich source of relevant scientific insights that can be exploited for the invention of novel visualization or interaction metaphors. In fact, various potentially interesting aspects discovered by cognitive science still have no practical counterpart in closely related fields, such as information visualization.
- **Motion.** The other main research theme investigates the viability of motion for visualizing information. The results and subsequent analysis have shown that the features of motion, as opposed to the traditional method of the object moving, can be employed for useful information display. In particular, motion is able to effectively represent dynamic data characteristics by two concurrent processes: the generation of different cognitively perceivable *behavior typologies* and the creation of *static artifacts* with distinct, interpretable visual features.
- **Self-Organization.** The still largely unexplored, yet powerful potential of *self-regulating rules* for information visualization purposes has been clearly demonstrated. The infoticle method has shown that the parallel execution of local behavior rules on a collection of individual entities is able to efficiently both process and represent large amounts of complex and time-varying abstract data in real time.

- **Value.** The infoticle metaphor is relevant to the problem of information visualization, because many time-varying datasets consist of data entities that are somehow conceptually interrelated, be it in terms of *data value*, *data value alteration*, *data value history* or *data update frequency*. The representation of these dynamic data characteristics is especially valuable and meaningful for at least three different purposes.
  - **Passive presentation** and **demonstration** of dynamic data evolutions to a non-expert audience.
  - **Data exploration** to detect unknown time-based features and data value characteristics.
  - **Detailed data analysis** to determine the exact attributes of known or expected data patterns.
- **Presentation Medium.** The infoticle method effectively exploits the potential of various unique virtual reality technology features. For instance, *stereoscopic vision* improves the effective perception of motion and drastically increases the amount of points that can be understood, whereas *physical immersion* enhances the feeling of presence and spatial orientation in relation to the dataset representation. The combination of arbitrary, real-time spatial navigation and direct manipulation techniques facilitates the comprehension of the relation between the visual data representation and the corresponding conceptual data structure.
- **Metaphor.** The visualization method represents differences in update characteristics by generating distinct visual patterns, while simultaneously offering a continuous contextual overview over the whole dataset. These patterns can be perceived either *dynamically* or *statically*, and on a *global* or *microscopic* scale. The main cognitive features that enable intuitive understanding of infoticle patterns include *pattern similarity, metaphor consistency* and *variation causality*.
- **Methodology.** The infoticle methodology explains in detail the adopted *design rationale*, and justifies particular design choices in the context of related insights originating from diverse disciplines. In addition, the theoretical description documents the employed logical and implementation procedures that form the conceptual basis of the visualization metaphor. It enables future visualization application designers to extend various aspects of the infoticle technique and allows for potential users to evaluate the expected validity of the infoticle methodology in the context of their own dataset and presentation medium.
- **Implementation.** Most system implementation issues transcend the proposed infoticle method to time-varying information visualization in general. In fact, many of the aspects described in this thesis are generally valid for *querying, communicating, processing, updating* and *simulating* large streams of time-varying datasets from remote data farms to local visualization applications in real time, regardless of the exact data mapping algorithms or the application purposes. These considerations also include performance measures to optimize the streaming and handling of data queries, the use of virtual reality technology and the implementation of intuitive interface paradigms.

- **Analysis.** Because of the originality and unfamiliarity of the generated visual patterns, the emergent *compositional* and *grammatical principles* have been evaluated. This analysis has several relevant aims: to be able to explain the basic principles of most emergent visual patterns, even if they are novel and unexpected; to justify particular design choices in the context of the problem of time-varying information visualization; to explain the pattern methodology to unfamiliar users; to derive generally valid time-dependent information visualization design principles; and to serve as a foundation model for future work in the area of both motion-based and emergent information visualization.
- **Guidelines.** As a direct outcome of the infoticle method analysis, a set of guidelines that transcend the infoticle realm has been developed.
  - **Consistency.** Breaking metaphor consistency is an effective technique to dynamically and instantaneously *alert* users, and to generate static *shape* irregularities in spatial artifacts.
  - **Adaptation.** The continuous reevaluation and equilibrium-reaching evolution of a dynamic representation can be exploited to cluster long-term time-based tendencies. In fact, the adaptation time span needed by the visualization system to stabilize the time-varying infoticle constellation into a true informational representation facilitates the faster clustering of data objects with equal *data alteration evolutions*.
  - **Causality.** The data update *event* itself, and not only the exact data value, can be used to generate a meaningful representation of various dynamic data characteristics, such as the relative update frequency. The resulting patterns need to be easily interpretable by users, following the principles of cognitive perception.
- **Motion.** Distinct motion typologies convey different informational meanings and enable effective perceptual data clustering by *speed*, *direction*, *behavior* and *temporal proximity*. Not the object moving, but the *nature of the movement* can be the driving force of a usable information visualization approach.
- **Update.** Time-varying datasets contain various data tendencies with valuable, informational values that should be visualized: exact *data value*, *relative data value change*, *data value history*, *data update frequency* and *data update history*.
- **Evaluation.** The infoticle method offers features that exceed those that can be found in other visualization techniques. Its most remarkable characteristics include the ability to: offer a continuous contextual overview over the whole dataset, to instantaneously process information by data-dependent interactions within the representation, to use behavior rules to control the visualization points, to merge spatial modeling with information processing, to exploit the nature of motion, to utilize artifact formality for data analysis, to base design decisions on cognitive science insights, to use a dynamic as well static representation for data interpretation, to gradually stream data in real time to the visualization engine, to implement parallel processing to guarantee motion continuity, to simulate timelines in different timeframe granularities and directions, to create global as well as local behavior patterns and to merge fundamentally different data dimensions within the same representation space. As a result, diverse data patterns have emerged that were not previously detected by other visualization approaches.

- **Computer Science.** The infoticle system description has demonstrated how various known techniques developed by computer science research that are normally used for real-world simulation and mimicking purposes, such as particle animation, spline calculation, surface generation and texture rendering, can be combined creatively to create novel visualization methodologies. This research proves how computer science can deliver several relatively novel and independent procedures for unanticipated but meaningful purposes.
- **Application.** The infoticle methodology is especially suited for early data exploration purposes, when little is known about the time-varying characteristics of the dataset. It broadens the possible recognition of simultaneous data patterns by breaking the fixation on static assumptions, and by using a novel metaphor, widens the hypothesis set to explain the available data. In effect, it draws the user's attention to highly informative dynamic relationships between time-varying data entries, even when the user does not know to look for that data explicitly and only was aware of its static features.

6.2. Future Work

This research has shown that data-driven particles are highly applicable to datasets with time-varying data values, and ideally should be used in immersive virtual reality environments. In addition, the analysis (see Chapter 5) has demonstrated the visualization method's potential in several different contextual situations.

- **Metaphor.** There is a general need to understand and appreciate what makes metaphors effective. In fact, the best metaphors typically combine creativity and experience, algorithms and aesthetics, have been gradually refined, and have stood the test of time. It is unanticipated whether the infoticle metaphor will receive a general acceptance, although this thesis has clearly shown the great potential of data-driven particles for information display. In the future, various evaluations and re-implementations need to be completed, so as to identify the most productive parameters and remove those factors that disturb the method's effectiveness. As a result, a reduced infoticle representation might emerge that demonstrates the core functionality of the metaphor. More complex infoticle representations might result that appeal more to the intuition and aesthetic feeling of users. For instance, the practical consequences of including dynamically moving forces and filters driven by the collective attractions of infoticle collections are still largely unknown. In addition, forces could represent multi-dimensional data conditions, most probably resulting in new, unpredictable dynamic behaviors.
- **Technology.** The proposed infoticle system reached the borders of today's technology in several contexts. Consequently, technological enhancements in interaction mechanisms (speech recognition, gesture recognition, etc.) and projection devices (increased screen resolution, larger displays, etc.) would offer new opportunities to improve the effectiveness of the infoticle method. Furthermore, various optimizations on the software code level, such as the implementation of more sophisticated particle systems (Carrard, 2001) or data processing algorithms would dramatically enhance the application framerate and thus the apparent smoothness of the timeline simulation.

- **Interface.** Because of the current experimental nature of the infoticle method, many influencing parameters are hard-coded within the software, making them practically inaccessible for ordinary users. In fact, one of the major limitations of the current infoticle system state is the inability to be reconfigured at runtime. Therefore, user-friendly interfaces need to be implemented that allow for the fine-tuning of the data mapping rules between specific thresholds and the adjusting of conditional tool values.
- **Dataset.** Some practical but powerful capabilities regarding dataset characteristics have still not been implemented, such as the visualization of real-time data, or the effective use of multi-dimensional forces. The simultaneous processing of several data dimensions most probably would result in more complex representations that are difficult to comprehend. Although numerous different dataset sources could be imagined to be visualized using the infoticle method, the following two scenarios deserve some additional attention because of their foreseeable business value.
  - **DNA Sequence.** Modern molecular biology is one of the research fields that witness an explosive growth of scientific data, and currently is in the process of developing genetic databases. These collections contain massive information about DNA sequences, the proteins encoded by them, and the homologies between different genes. It is one of the academic fields that need to explore tremendous amounts of data in search for highly valuable data patterns, yet are still unaware how to reach this goal most effectively. Infoticles could represent some DNA sequence analysis attributes, such as the evolution of genome expressions over long periods of time.
  - **Project Management.** The financial day-to-day overviews of many companies, which track systematically which employee works on what project and when, could be another valuable dataset. Symbolizing projects by forces and employees by infoticles would allow for the detection of repeating or problematic tendencies. Managers would typically show up as quarks, encompassing various projects at once, whereas employees dedicated to single projects would pop-up as electrons or comets.
- **Grid.** Grids are complex distributed systems that facilitate the access of computational resources in a transparent fashion, and provide an infrastructure for computationally intensive research in, for instance, particle physics, molecular biology or earth sciences. At the hearth of the grid concept lies the collaborative handling of massive datasets, and the visualization of patterns that are hidden within (Foster, et al., 1999). The infoticle method could provide one possible solution to both problems: it facilitates the visualization of data that needs to be gradually streamed to the system because of its shear size, and enables the organization of this data according to various, even time-varying, characteristics.
- **Presentation Medium.** A great potential still lays ahead in the evaluation of the infoticle method applied on other presentation media, such as traditional two-dimensional desktop displays, web-based interfaces or small devices such as Personal Digital Assistants (PDAs). These new presentation and interaction environments would inevitably require the infoticle metaphor to be reduced to its core functionality, so that it can be executed on simple calculation and rendering hardware and displayed on low resolution displays.
- **Collaboration.** Although initially considered (see Section 3.6.4. Infoticle – Interface – Collaboration), the collaboration possibilities of a shared infoticle world never have been tried out in real-world circumstances. Conceptually, remote participants could share a common infoticle simulation, each perceiving the same spatial particle

constellations with different, personalized data attributes. Such a collaboration scenario would extensively rely on blue-c's unique, real-time human reconstruction capabilities, because three-dimensional gesturing is fundamentally important when working with the dynamic and spatially dependent infoticle artifacts within three-dimensional space. However, several software measures would need to be implemented first in order to avoid a continuous electronic transfer of the large amount of points. Instead, a possible solution would consist of synchronizing user interactions and update frequencies for all participants, out which the infoticle constellation changes would be independently calculated at the remote sites.

- **Agents.** Especially boid infoticles have shown considerable similarity with the concept of agents, in that they are aware of their informational meaning and their surrounding environment. At least in theory, each boid infoticle could be extended with some significant purpose, such as the empirically derived rules of good visualization and the Gestalt rules. Consequently, animated *agent infoticles* would behave in a goal-oriented way, producing graphical patterns that users are able to perceive efficiently and understand cognitively. Possible applications could then feed raw data junks to such an agent system which, dependent on contextual parameters detected within the dynamic dataset and environmental influences surrounding the used presentation medium, is always capable to adapt and represent these effectively.
- **Future Prediction.** In theory, forces have the ability to continuously trace the amounts and types of infoticles they have attracted in time. As a result, this empirically gathered experience could lead to a workable infoticle visualization system that simulates data tendencies *in the future*. Practically, by extrapolating the different force influences, the continuous interplay of infoticles and forces can be extended in time without any available data entries within the dataset. Furthermore, one could alter these force strengths according to possible foreseeable data evolution scenarios to analyze the resulting time-varying patterns. Alternatively, each single infoticle could also track its behavior in relation to the simulated timeline, and interpolate this historical experience in the future as well. Possible application areas include the prediction and simulation of 'what-would-happen-if' scenarios, such as financial datasets in which distribution matrices have changed in direct relation to well-known variables of the years before.

6.3. Conclusion

In summary, the first part of the thesis has described the main motivations and conceptual basis for the design of a novel explorative information visualization metaphor that exploits motion typology generation and immersive virtual technology to convey time-varying data. Subsequently, the core principles and implementation issues of the visualization system have been discussed in detail, and have been justified in the context of scientific insights from diverse related disciplines. The last part of this research has proven the validity and flexibility of the data-driven particle concept for visualizing abstract information, by the development and subsequent analysis of several application prototypes that represent various real-world dynamic datasets.

A. blue-c

The presented work was carried out within the context of the blue-c project. This project created a novel hard- and software system that successfully combines the advantages of a CAVE-like projection environment with simultaneous real-time three-dimensional video capturing and processing of the user. As a major technical achievement, users can now become part of the visualized scene while keeping visual contact with each other. Consequently, these features make the system a powerful tool for high-end remote collaboration and presentation. Two portals have currently been implemented with complementary characteristics, and are networked with a gigabit connection. One portal is located at the ETH main campus, the second at ETH Honggerberg. Various applications have proven the concept and demonstrated the usefulness of the blue-c technology.

Figure A.1 shows the ETH main campus blue-c portal in action. It is a three-sided CAVE-like portal with actively shuttered projection walls. The second blue-c installation located at ETH Honggerberg is a single projection wall setup. Therefore this installation requires less demanding solutions for the simultaneous projection and acquisition than the first installation. Both systems acquire the three-dimensional video inlay with 16 cameras and have stereo projection capabilities.

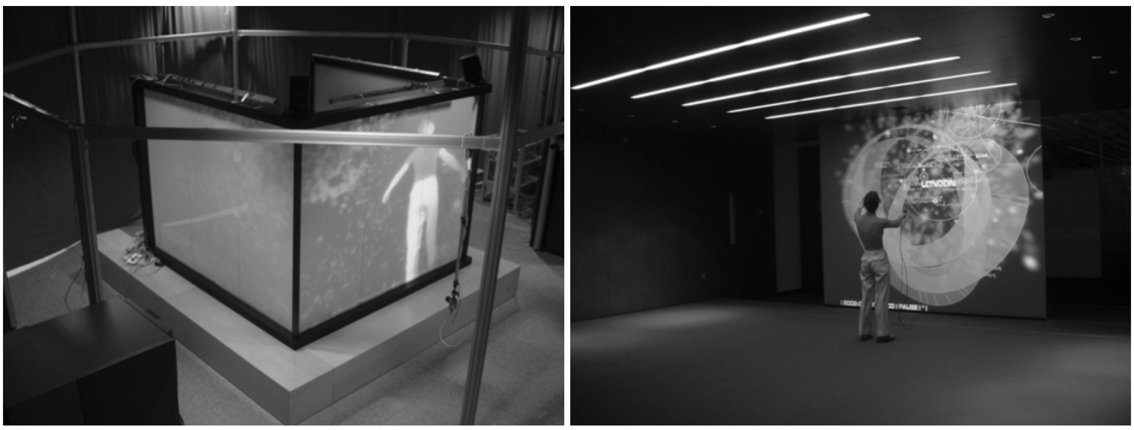


Figure A.1. blue-c portals.

The blue-c project was organized as an ETH internal research project. It started on May 1, 2000 and lasted for three years. Four ETH research groups participated in this project, namely the Computer Graphics Lab (CGL) under the supervision of Prof. Markus Gross, the Computer Vision Laboratory (CVL) under the supervision of Prof. Luc Van Gool, the Center of Product Development (ZPE), which is part of the Institute of Mechanical Systems (IMES) under the supervision of Prof. Markus Meier and the chair of Computer Aided Architectural Design (CAAD) supervised by Prof. Maia Engeli at the start of the project and later on by Prof. Ludger Hovestadt.

The CGL directed the project and was responsible for the core software components, including graphics rendering and three-dimensional video processing, as well as for the computing and networking infrastructure. The CVL took care of the silhouette extraction from the captured images and of the camera calibration. The ZPE was responsible for the hardware and projection setup, including the construction of the first blue-c portal at ETH main campus. The CAAD chair investigated applications and interaction techniques, designed the virtual reality installations and built the second blue-c portal at ETH Honggerberg.

Figure A.2. illustrates the framework of the core components of the blue-c project. These core components are briefly described in the following paragraphs.

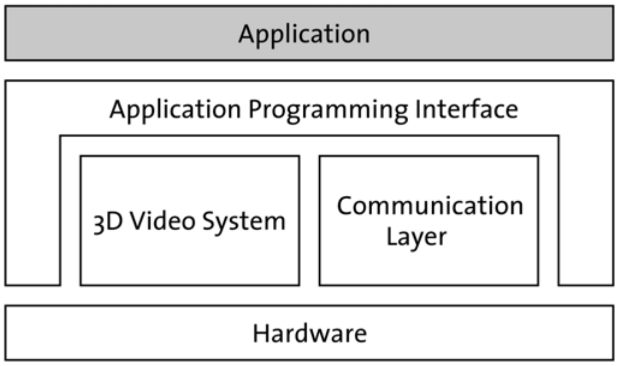


Figure A.2. blue-c framework

- **Hardware.** The hardware includes the projection system, the integration of the acquisition system, the illumination and the synchronization electronics for all components. The hardware has to satisfy the contradicting needs of projection and image acquisition. The blue-c hardware is the subject of the thesis of Christian Spagno (2003).
- **3D Video System.** The 3D video software computes a 3D representation of the acquired persons from within a blue-c installation in real time. The software first builds a 3D point cloud from multiple camera video streams. The point cloud can then be efficiently encoded and streamed to a remotely located blue-c portal where it is rendered into the application. The thesis of Stephan Würmlin deals with the 3D video system (to appear).

- **Communication Layer.** The acquisition and rendering systems of the distributed installations are interconnected and thus enable for tele-collaboration applications. Strict temporal and bandwidth constraints have to be met in order to enable high quality collaboration with live video streams. Furthermore the system needs to adapt to changing qualities of the network services. The communication layer is detailed in Edouard Lamboray's thesis (to appear).
- **Application Programming Interface.** The blue-c Application Programming Interface (blue-c API) exposes the blue-c system functionality to the application developer. It provides a rapid application development environment that supports tele-collaboration and the integration of multimedia data into the virtual world. The blue-c Application Programming Interface is the subject of Martin Naef's thesis (to appear).
- **Application.** The applications run on top of the blue-c API and are customized to the requirements of the specific blue-c hardware and software layout in order to fully exploit the full potential of this new system. Two application areas have been researched in the framework of the blue-c project. Infoticles is a novel information visualization metaphor that uses the motion characteristics of particles to explore unexpected data patterns in large, time-varying datasets. This visualization technique is described in detail in this thesis. IN:SHOP is the first application to investigate and analyze the possibilities of integrating the blue-c technology into buildings. Therefore, a novel approach to distributed shopping in introduced, by extending physical shopping floors into virtual and remote spaces. The impact of video stream systems on architecture is described in detail in the thesis of Silke Lang (to appear).

B. Glossary

- **Agent**. A software object with a certain amount of autonomy in the sense that it has some control over its actions and internal state. An agent is able to perceive its environment, and responds to the changes that occur therein in a goal-directed way.
- **Ambient Display**. An abstract and aesthetic peripheral display portraying non-critical information on the periphery of a user's attention (Wisneski, et al., 1998).
- **Application Timeframe**. A time duration, mostly measured in a number of frames, during which the application visualizes data entries of a single database timeframe. After each application timeframe, the system stores the current state, and new data is fetched for the next database timeframe.
- **Application Protocol Interface (API)**. A software interface that is used for accessing an application or a service by a program. An API makes it possible to use services within programs. Therefore it forms the foundation for modular systems with clearly defined interfaces between separate components.
- **Apprehension Principle**. The structure and content of the external representation should be readily and accurately perceived and comprehended (Tversky, et al., 2002).
- **Atomic Initialization**. A data object-infoticle mapping mechanism, characterized by a singular relationship between database and infoticle. During initialization, each database entry within the initial database timeframe corresponds to a single infoticle.
- **Behavior**. A certain way of acting as a result of conditioned responses.
- **Blob**. Something of vague or indefinite form. Also described as organic, sticky, viscous, mobile composite entity capable of incorporating disparate external elements into itself.
- **Browsing**. To explore or to scan through a media collection without a focused goal.
- **Brushing**. An interaction process that enables the user to highlight, select or delete a subset of elements by pointing to the elements with a pointing device (Wills, 1996).

- **Cognition**. A way of looking at human behavior that emphasizes how the brain takes in information, creates perceptions, forms and retrieves memories, processes information, and generates integrated patterns of action. This process includes insights from both awareness and judgment.
- **Cognitive Ability**. The capacities to reason, remember, understand, solve problems and make decisions.
- **Cognitive Distance**. The distance between human thoughts and the physical requirements of the system under use.
- **Cognitive Psychology**. The study of the mental processes by which information from the environment is modified, made meaningful, stored, retrieved, used and communicated to others.
- **Congruence Principle**. The structure and content of the external representation should correspond to the desired structure and content of the internal representation (Tversky, et al., 2002).
- **Consistency**. The quality of preserving the same spatial relations and underlying notions across the representation and interaction inside a virtual world and its chosen visualization metaphor.
- **Cue-of-Motion**. Also called structure-from-motion. When an object or an observer moves, the objects located at different distances move with different speeds and create patterns of motion parallax and kinetic depth that provide highly informative depth cues.
- **Cyber**. Etymologically, cyber means steersman, whereas today, it connotes automation, artificial control and computerization.
- **Cyberspace**. Term coined in 1984 by science-fiction writer William Gibson (1984), denoting a multi-user virtual world built out of data constructs. Currently cyberspace has many meanings.
- **Data**. Information in a raw form that can be digitally transmitted or processed.
- **Data Attribute**. A characteristic common to all or most instances of a particular entity, typically used to assign corresponding data values. Synonyms include property, data element, identity and field.
- **Database**. A collection of information in the form of a set of individual entries, along with a set of links between pairs of entries that indicate some particular relationship between them.
- **Database Timeframe**. A specific timeframe duration, mostly measured in physical time units, in which the time identifications of a subset of the data stored in the database are situated. Normally, the data within a specific database timeframe is visualized during a single application timeframe.
- **Data-Driven Particle**. See Infoticle.
- **Data Object**. A uniquely identifiable collection of data attributes with according data values. Synonyms include *instance*, *identifier*, and *primary key*. For instance, a single person could be a data object with name and address as constant data attributes. The exact street name and number are data values that can change in time.

- **Dataset**. A series of data objects that are originally created or managed on a computer system. A dataset usually takes the form of fields and tables within which items of data are contained and structured.
- **Data Update**. The change of data values to a successive database timeframe of a data object, which is uniquely identified by a data object identifier. Happens on the rhythm determined by the application timeframe.
- **Data Update Frequency**. The frequency of data updates. Also: the frequency with which a data object is subjected to alterations during a single application timeframe.
- **Data Update History**. A list of the historical tendencies related to the characteristics generated by a sequence of data updates. Such list might consist of the historical values of the data update frequencies and data value updates. It is not necessarily stored in a computer's memory, but might be mnemonically memorized by visual artifacts that trace the data update sequences.
- **Data Value**. The exact numerical or nominal value of a data attribute.
- **Data Value Change**. The relative change of data object values after a data update. Data value changes can be compared with the rest of the dataset for that database timeframe and be characterized e.g. as significant, relatively large, etc.
- **Data Value History**. A list of the historical data values that were changed as a result of a sequence of data updates. Similar data value histories contain similar data values in time.
- **Data Value Update**. The occurrence of data values changing to the active database timeframe. See also Data Value Change.
- **Demo-Oriented Research**. Research based upon the implementation of concrete, real-world working prototypes to demonstrate the foreseen capabilities and evaluate the final outcome, meanwhile learning from the development process.
- **Design Rationale**. The description of why a certain design is the way it is. What motivated a particular design, what alternatives were considered and rejected, etc.
- **Electron**. A negatively charged particle making up the outer shell of an atom.
- **Emergent Behavior**. A global effect generated by local rules that arises because of interaction between subunits in the system. Consequently, emergent behavior cannot be predicted through analysis at any level simpler than that of the system as a whole.
- **Equilibrium**. Some balance occurring in a model, which can represent a prediction if the model has a real-world analogue.
- **Equilibrium Adaptation Time.** The time duration required by the visualization system to reach a true, stable representation. Consequently, a true representation can be analyzed without the risk of misinterpretation. In practice, the application timeframe is mostly shorter than the required equilibrium adaptation time, bringing the resulting representation under a continuous stress.
- **Exabyte**. A large unit of computer data storage, equal to 260 bytes. In decimal terms, an exabyte is a billion gigabytes.

- **Framerate**. Defines how many computer-generated pictures or frames are rendered within one second. The acronym of framerate is *refresh rate* or *frames per second* (fps). This variable is especially important in real-time graphics, which typically requires sophisticated graphics software to keep the framerate above the threshold at which the human eye can perceive picture changes (e.g. flicker) instead of smooth transitions.
- **Glyph**. A geometrical object with information encoded in the geometry or associated attributes such as color, size, etc.
- **Grammar**. Set of rules for combining the words used in a given language.
- **Human-Computer Interaction (HCI**). The exchange of information between human beings and computers during a task sequence for the purpose of controlling the computer (from the point of view of the human) or informing the user (from the point of view of the computer). This interaction usually aims at increasing human productivity, satisfaction, or ability (Hix and Hartson, 1993).
- **Immersion**. The feeling of being within an environment. Virtual reality technology immerses a user physically, whereas most media creators try to immerse users mainly mentally. Virtual reality users are immersed when they feel that the virtual world surrounds them and has, to some degree, replaced the physical world as the frame of reference. Immersion may take place in other media, such as films or even books (Bowman, 1999).
- **Immersive**. Surrounding the user in space. A computer-generated environment is described as immersive when it appears to enclose the user, and when parts of the physical world that are not integral system components are blocked from view (Bowman, 1999).
- **Implicit Rendering**. The rendering of implicit surfaces. Implicit surfaces are contours or isosurfaces through some scalar field in three-dimensional space.
- **Information.** Data with given meaning by way of relational connection
- **Information Exploration**. A conceptual approach that employs mostly graphical techniques to maximize insight in the dataset and to test underlying assumptions.
- **Informative Art.** Specially designed, computer-augmented or amplified, works of art that are not only aesthetical objects but also information displays, in as much as they dynamically reflect information about their environment by manipulating known art styles or developing ambient visual metaphors (Holmquist and Skog, 2003)
- **Infoticle**. Synonym for data-driven particle, as a particle that represents an abstract data object. The corresponding particle attributes, such as speed, color, transparency, lifespan, direction and acceleration, depend on the actual data values it represents. Due to the continuous update of these data values, data update characteristics become visualized by emergent visual patterns, such as dynamic behavior typologies, spatial clusters and three-dimensional shapes.
- **Interaction Technique (IT)**. A method by which the user performs a task on a computer via the user interface.
- **Intranet**. An internal network architecture based on Internet standards. Users access it through the same browser software that they use to surf websites on the Internet.
- **Jovicentric**. Revolving around a point, not necessarily in a stable trajectory.

- **Multiplied Initialization**. A data interpretation approach resulting in a one-to-many relationship between a single database entry and a set of corresponding infoticles. In practice, this means that a specific quantifiable data value within a database row is divided into a discrete amount of corresponding infoticles.
- **Navigation**. The process of determining a path to be traveled by any object through any environment.
- **Parallel Sequential Dataset**. A dynamic dataset with a one-to-many or one-to-null relationship between a specific database timeframe and a unique data object. In practice, this means that multiple data values for a single identifiable data object are available within a specific database timeframe. Mostly, this frequency issue is solved by averaging or counting those data values for each database timeframe.
- **Perception**. The process by which people take raw sensations from the environment and interpret them, using knowledge, experience, and understanding of the world, so that the sensations become meaningful experiences.
- **Pseudocode**. An outline of a program, written in a form that can easily be converted into real programming statements. Pseudocode cannot be compiled nor executed, and there are no real formatting or syntax rules.
- **Psychophysics**. Partly physical, partly psychological, psychophysics denotes all types of investigations into the nature of sensory processes, especially vision. The term applies to the vision process that is concerned with the transfer of information from the eyes to the effectors, either directly or through the medium of memory.
- **Pre-Attentive**. A cognitive principle that denotes the preconscious as opposed to deliberate processing of information. In general, anything that is processed at a faster rate than 250msec per item is considered to be pre-attentive.
- **Presence**. A synonym for immersion. Also used to denote 'sense of presence', i.e. being mentally immersed, deeply engaged and involved.
- **Proprioception**. A person's sense of the position and orientation of the body and its parts.
- **Quark**. A fundamental particle. Six types of quarks are known. Up and down flavors are constituents of protons and neutrons. The other, heavier, quarks are called strange, charm, bottom and top.
- **Query**. The process of selecting a subset of entries, possibly maintaining the knowledge of all links involving members of that subset. If the user has a definite target, the process is called *querying*. If the user has a less specific target and some interest in serendipity, the process is called *browsing* or *exploring*.
- **Scene**. Synonym for a complete virtual world, consisting of a set of graphic objects and a navigation. A navigation is a viewpoint and viewing direction evolving over time. A scene graph also denotes a hierarchical and ordered list of all elements in the virtual world which enables both the designer and the computer system to handle the interactions, animations and rendering of that world in an efficient way.
- **Semiotics**. A field stat studies signs and how meanings are constructed.
- **Simulation Timeframe**. See Application Timeframe.

- **Singular Sequential Dataset**. A dataset consisting of data objects that change only once within each database timeframe, resulting in a one-to-one relationship between each data object and each database timeframe and application timeframe.
- **Syntax**. The study of the rules governing sentence structure, the way words work together to make up a sentence.
- **Tele-Immersion**. The ability to let users see and interact with each other and objects inside the shared virtual environment in real time. The physical representations of the participants (instead of avatars) are ported within a shared, virtual environment.
- **Telepresence**. The ability to directly interact with a physically real, remote environment from a first-person point of view (Sherman and Craig, 2003).
- **Time-Varying Data.** Also *temporal data*, *time-dependent data*, *time-based data* or *dynamic data*. A dataset consisting of uniquely identifiable data objects that reoccur in time with changed data values.
- **True Representation**. A representation in which the visual cues of all elements present in the scene truly and flawlessly denote their corresponding informational values. Mostly a system needs a specific adaptation time to reach this state after the data values have changed due to a data update.
- **Update**. Change, invoked by a successive shift in time.
- **Update History**. The evolution of an accumulated sequence of data updates. The relative qualitative change of a single data update can be judged either in comparison with parallel updates of other similar data objects or with the time update history of the same data object.
- **Usability**. The relationship between tools and their users. In order for a tool to be effective, it must allow intended users to accomplish their tasks in the best way possible. In order for these systems to work, their users must be able to employ them effectively. Usability usually refers to software but is relevant to any product. Some ways to improve usability include: shortening the time to accomplish tasks, reducing the number of mistakes made, reducing learning time, and improving people's satisfaction.
- **Usage**. The study that deals with the issues of how tools can be used.
- **User Interface (UI**). The hardware and software that mediate the interaction between humans and computers. The UI includes input and output devices, such as mice, keyboards, monitors, and speakers, as well as software entities such as menus, windows, toolbars, etc. (Hix and Hartson, 1993).
- **Virtual Reality**. In the context of this thesis, virtual reality denotes Immersive Virtual Reality (IVR), or CAVE-like environments, in which the user is physically surrounded by the computer-generated virtual representation.
- **Wand**. A Six-degree Of Freedom (6-DOF) input device typically used for virtual reality environments, which is able to measure the exact spatial position and orientation within physical space. Often, a wand has extra buttons and a joystick built-in.

C. Color

Figure C.1. Modeling World.

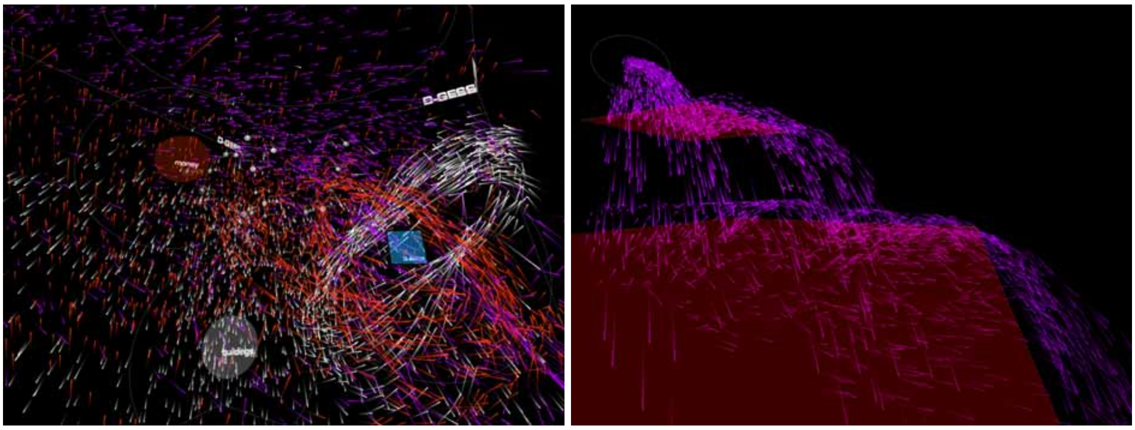


Figure C.2. Modeling World patterns.

Figure C.3. Galaxy World collaboration

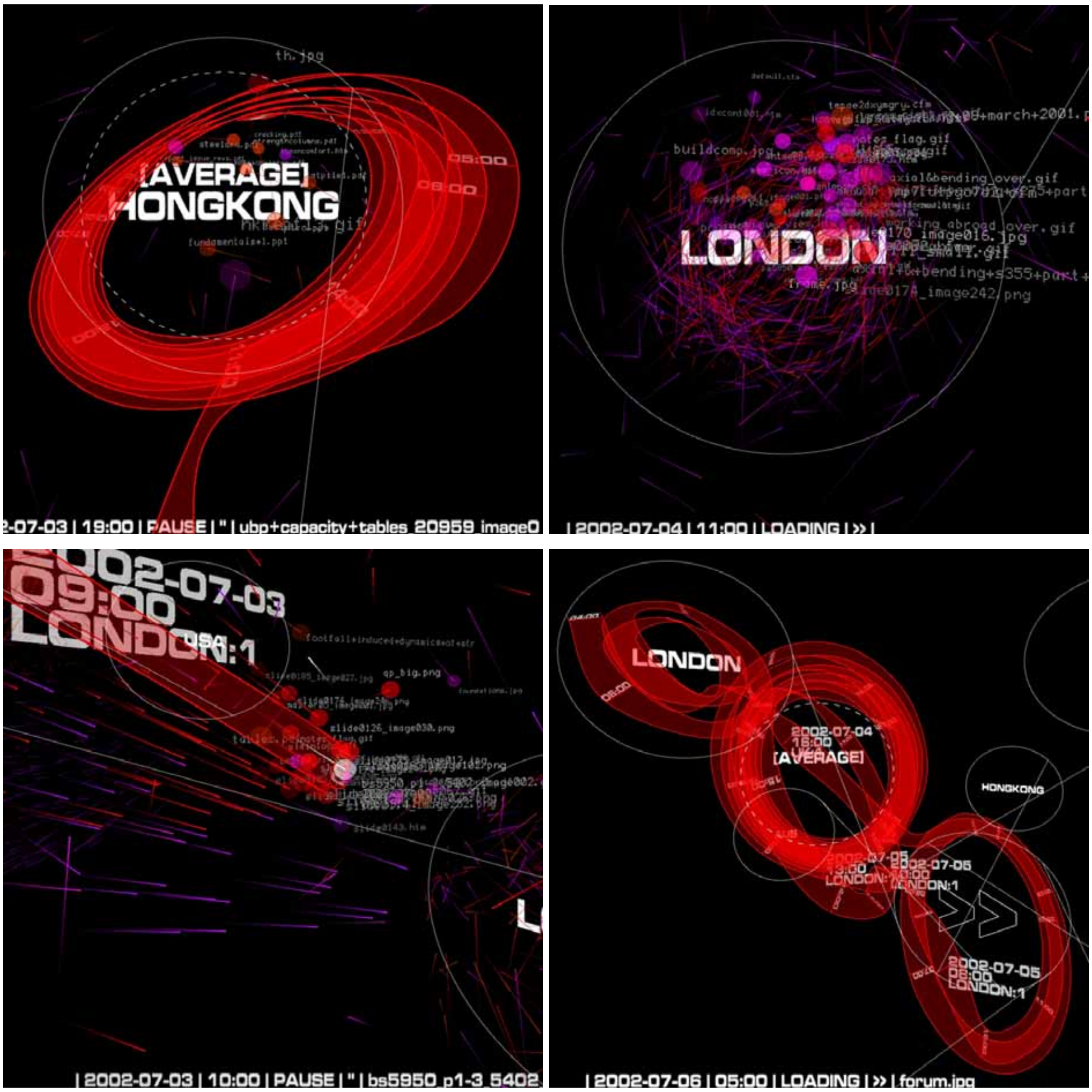


Figure C.4. Galaxy World patterns.

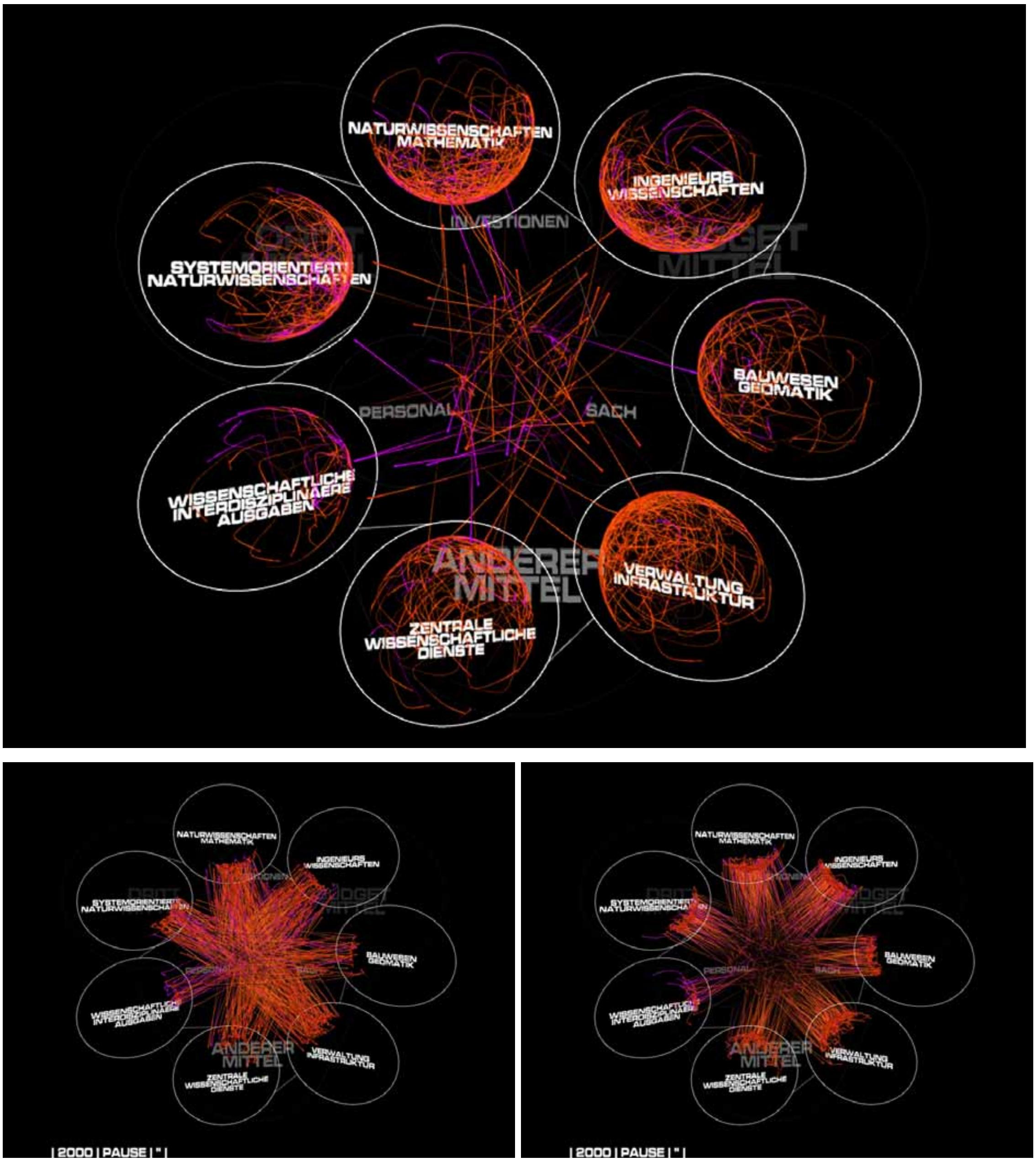


Figure C.5. Electron World time and data dimension alterations.

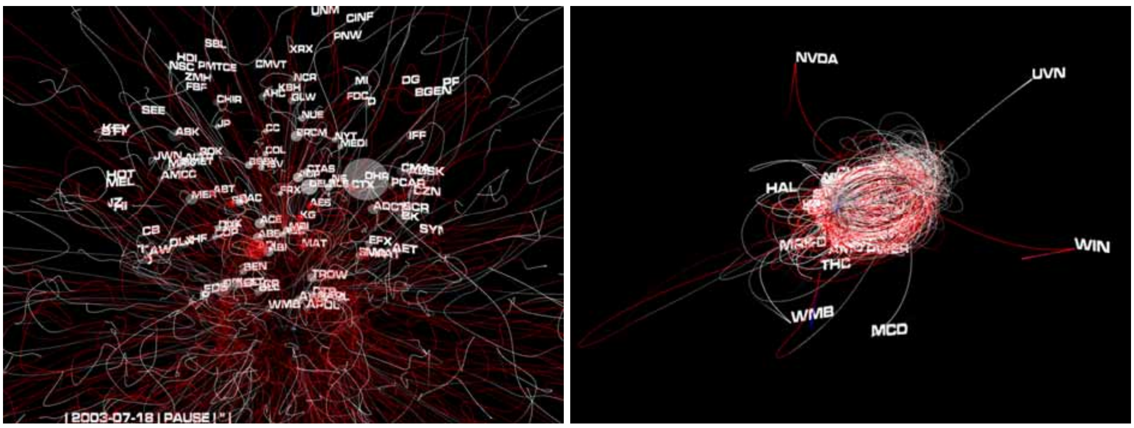


Figure C.6. Boid World.

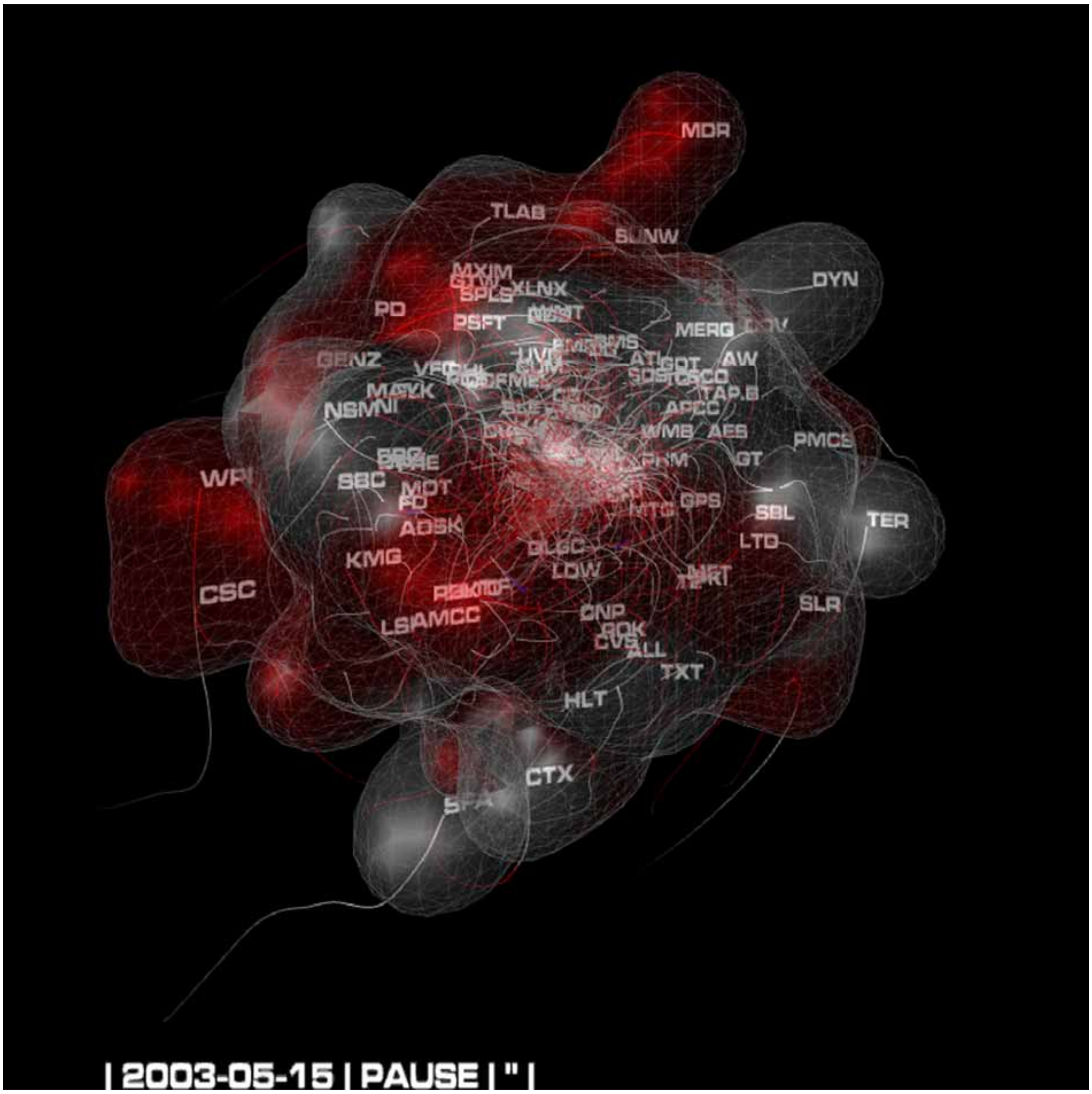


Figure C.6. Boid shape.

D. References

Abel, P., Gros, P., Russo Dos Santos, C., Loisel, D. and Paris, J. P. (2000) *Automatic Construction of Dynamic 3d Metaphoric Worlds: An Application to Network Management*, Visual Data Exploration and Analysis VI, SPIE - The International Society for Optical Engineering, pp.312-323.

Adams, R. M., Stancampiano, B., McKenna, M. and Small, D. (2002) *Case Study: A Virtual Environment for Genomic Data Visualization*, IEEE Visualization 2002 (Vis'02), IEEE, Boston, MA, pp.513-516.

Ammoura, A. (2001) *Dive-On: From Databases to Virtual Reality*, ACM Crossroads (7) 3, pp.4-ff.

Anders, P. (1999) *Envisioning Cyberspace,* McGraw-Hill, New York.

Arns, L., Cook, D. and Cruz-Neira, C. (1999) *The Benefits of Statistical Visualization in an Immersive Environment*, Virtual Reality, IEEE, pp.88-95.

Art+Com (2003) *Art + Com Ag,* [Online], Available from: http://www.artcom.de, [Accessed: October, 2003].

Azzopardi, L. (1999) *Wave Motion: A New Metaphor for 2d Information Visualization*, Pan-Sydney Area Workshop on Visual Information Processing, Sydney, Australia.

Babski, C. and Carion, S. (2003) *_Knowscape - a Collective Knowledge Architecture*, VECIMS 2003 - International Symposium on Virtual Environments, Human-Computer Interfaces, and Measurement Systems, Lugano, Switzerland.

Badler, N. I. and Tsotsos, J. K. (Eds.) (1986) *Motion Representation and Perception Proceedings of the Acm Siggraph/Sigart Interdisciplinary Workshop (Toronto 1983),* North-Holland, New York a.o.

Baecker, R. and Small, I. (1991) Animation at the Interface, In *Computers as Theatre* (Ed, Laurel, B.), Addison-Wesley, Reading, Massachusetts etc., pp.251-267.

Bartram, L. (2001) *Enhancing Information Visualization with Motion,* [Thesis], School of Computing Science, Simon Fraser University.

Bartram, L. and Ware, C. (2002) *Filtering and Brushing with Motion*, Journal of Information Visualization (1) 1, pp.66-79.

BBC (2003) *Black Holes*, In Space - with Sam Neill, BBC Television, London, UK.

Becker, R. A., Eick, S. G. and Wilks, A. R. (1995) *Visualizing Network Data*, IEEE Transactions on Visualization and Computer Graphics (1) 1, pp.16-28.

Benedikt, M. (1991) *Cyberspace: First Steps,* MIT Press, London.

Benford, S., Snowdon, D., Greenhalgh, C., Ingram, R., Knox, I. and Brown, C. (1995) *Vr-Vibe - a Virtual Environment for Cooperative Information-Retrieval*, Eurographics 95, Maastricht, The Netherlands, pp.349-360.

Bloomenthal, J. (1988) *Polygonization of Implicit Surfaces*, Computer Aided Geometric Design (5), pp.341-355.

Bloomenthal, J. (1994) An Implicit Surface Polygonizer, In *Graphics Gems Iv* (Ed, Heckbert, P.), Academic Press, pp.324-349.

Bovey, J., Rodgers, P. and Benoy, F. (2003) *Movement as an Aid to Understanding Graphs*, International Conference on Information Visualization (IV'03), IEEE, London, United Kingdom, pp.472-478.

Bowman, A. D. (1999) *Interaction Techniques for Common Tasks in Immersive Virtual Environments,* [Thesis], Georgia Institute of Technology.

Bowman, A. D., Kruiiff, E., LaViola, J. J. and Poupyrev, I. (2001) *An Introduction to 3-D User Interface Design*, Presence Journal, Massachusetts of Technology (10) 1, pp.96-108.

Brodbeck, D., Chalmers, M., Lunzer, A. and Cotture, P. (1997) *Domesticating Bead: Adapting an Information Visualization System to a Financial Institution*, IEEE Symposium on Information Visualization (InfoVis '97), Phoenix, AZ.

Brookhaven (2003) *Rhic - Realistic Heavy Ion Collider,* [Online], Available from: http://www.bnl.gov/rhic/full_en_images.htm, [Accessed: October, 2003].

Bryson, S. (1992) *Virtual Spacetime: An Environment for the Visualization of Curved Spacetimes Via Geodesic Flows*, IEEE Visualization, ACM, Boston, Massachusetts.

Bryson, S. and Levit, C. (1991) *The Virtual Windtunnel: An Environment for the Exploration of Three-Dimensional Unsteady Flows*, IEEE Visualization, IEEE, San Diego, California, pp.17-24.

Bugajska, M. (2003) *Spatial Visualization of Abstract Information - a Classification Model for Visual Design Guidelines in the Digital Domain,* [Thesis], Department of Architecture, ETH-Zurich.

Bugajska, M. and Vande Moere, A. (2001) *Virtual Library - Paths to Knowledge*, Second International Workshop on User Interfaces to Data Intensive Systems (UIDIS'01), Zurich, Switzerland, pp.116-126.

Caplinger, M. (1986) *Graphical Database Browsing*, Conference Office Information Systems (ACM SIGOIS), Providence, RI, pp.113-121.

Carrard, P. (2001) *Eth World Conceptual Competition Virtual and Physical Presence,* gta Verlag, Zurich.

Carroll, J. M., Mack, R. L. and Kellogg, W. A. (1990) Interface Metaphors and User Interface Design, In *Handbook of Human-Computer Interaction* (Ed, Helander, M.), North-Holland, Amsterdam.

Chalmers, M. (1999) Informatics, Architecture and Language, In *Social Navigation in Information Space* (Eds, Munro, A., Hook, K. and Benyon, D.), Springer.

Chalmers, M. and Chitson, P. (1992) *Bead: Explorations in Information Visualization*, ACM Conference on Research and Development in Information Retrieval (ACM SIGIR), ACM Press, Copenhagen, Denmark, pp.330-337.

Chapman, P., Selvarajah, S. and Webster, J. (1999) *Engagement in Multimedia Training Systems*, Thirty-Second Annual Hawaii International Conference on System Sciences, IEEE, Maui, Hawaii, pp.1084.

Chen, C. (1999) *Information Visualisation and Virtual Environments,* Springer, London.

Chuah, M. (2002) *Demystifying Venture Capital Investing*, IEEE Symposium on Information Visualization (InfoVis'02), Boston, Massachusetts, USA, pp.161-165.

Cockburn, A. and McKenzie, B. (2002) *Evaluating the Effectiveness of Spatial Memory in 2d and 3d Physical and Virtual Environments*, SIGCHI Conference on Human Factors and Computing Systems, ACM Press, Minneapolis, Minnesota, USA, pp.203-210.

Crossno, P. and Angel, E. (1999) *Visual Debugging of Visualization Software: A Case Study for Particle Systems*, Visualization (VIS'99), IEEE Press, San Francisco, California, United States, pp.417- 420.

Cruz-Neira, C., Sandin, D. J. and DeFanti, T. (1993) *Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the Cave*, ACM SIGGRAPH, Anaheim, California, pp.135-142.

Cugini, J. V. and Scholtz, J. (1999) *Visvip: 3d Visualization of Paths through Web Sites*, International Workshop on Database & Expert Systems Applications, pp.259-263.

Dahley, A., Wisneski, C. and Ishii, H. (1998) *Water Lamp and Pinwheels: Ambient Projection of Digital Information into Architectural Space*, Conference on Human Factors and Computing Systems (CHI'98), ACM, Los Angeles, California, United States, pp.269-270.

Davis, B. J. (1998) *An Interview with Richard Saul Wurman*, Design Matters (3) 2.

Deyang, S. and Norman, M. I. (1994) *Looking in, Looking Out: Exploring Multiscale Data with Virtual Reality*, IEEE Computational Science and Engineering (1) 3, pp.53-64.

Dieberger, A. and Frank, A. U. (1998) *A City Metaphor to Support Navigation in Complex Information Spaces*, Journal of Visual Languages and Computing (9) 6, pp.597-622.

Dillon, A. (2002) *Information Architecture in Jasist: Just Where Did We Come From?*, Journal of the American Society for Information Science and Technology (53) 10, pp.821-823.

Douglas, Y. and Hargadon, A. (2000) *The Pleasure Principle: Immersion, Engagement, Flow*, ACM Conference on Hypertext and Hypermedia, ACM Press, San Antonio, Texas, United States, pp.153-160.

Drutt, M., Rashid, H. and Couture, L. A. (2002) *Guggenheim Virtual Museum,* [Online], Available from: http://www.guggenheim.org/exhibitions/virtual/virtual_museum.html, [Accessed: August, 2003].

Eades, P. (1984) *A Heuristic for Graph Drawing*, Congressus Numerantium (42), pp.149-160.

Earl, R. (1994) *Galaxy of News: An Approach to Visualizing and Understanding Expansive News Landscapes*, ACM symposium on User interface software and technology, ACM Press, Marina del Rey, California, United States, pp.3-12.

Eberhart, R. C. and Shi, Y. (2001) *Particle Swarm Optimization: Developments, Applications and Resources*, Evolutionary Computation, IEEE, pp.81-86.

Ebert, D. S., Shaw, C. D., Zwa, A., Miller, E. L. and Roberts, D. A. (1996) *Minimally-Immersive Interactive Volumetric Information Visualization*, IEEE Symposium on Information Visualization (INFOVIS '96), San Francisco, CA.

Eick, S. G. (2001) *Visualizing Online Activity*, Communications of the ACM (44) 8, pp.45-50.

Engeli, M. (2001) *Bits and Spaces: Architecture and Computing for Physical, Virtual, Hybrid Realms 33 - Projects by Architecture and Caad, Eth Zurich,* Birkhauser, Basel.

Erickson, T. D. (1990) Working with Interface Metaphors, In *The Art of Hci Design* (Ed, Laurel, B.), Addison-Wesley.

Fields, C. and Negroponte, N. (1977) *Using New Clues to Find Data*, Third International Conference on Very Large Data Bases, IEEE-CS and ACM, Tokyo, Japan, pp.156-158.

Foster, I., Insley, J., von Laszewski, G., Kesselman, C. and Thiebaux, M. (1999) *Distance Visualization: Data Exploration on the Grid*, IEEE Computer December 1999, pp.36-43.

Freeman, C. and Gelernter, D. (1996) *Lifestreams: A Storage Model for Personal Data*, SIGMOD Record (ACM Special Interest Group on Management of Data) (25) 1, pp.80-86.

Fuhrmann, A. and Gröller, E. (1998) *Real-Time Techniques for 3d Flow Visualization*, IEEE Visualization, IEEE Computer Society Press, Research Triangle Park, North Carolina, United States, pp.305-312.

Gibson, W. (1984) *Neuromancer,* Ace Books.

Grady, N. W., Tufano, D. R. and Flanery, R. E. (1998) *Immersive Visualization for Link Analysis*, AAAI Fall Symposium, Orlando, Florida, pp.98-101.

Gronquist, J. F., Sethares, W. A., Alvarado, F. L. and Lasseter, R. H. (1996) *Animated Vectors for Visualization of Power System Phenomena*, IEEE Transactions on Power Systems (11) 1, pp.267-273.

Gross, M. (1994) *Visual Computing - the Integration of Computer Graphics, Visual Perception and Imaging,* Springer Verlag, Berlin.

Gross, M., Würmlin, S., Naef, M., Lamboray, E., Spagno, C., Kunz, A., Koller-Meier, E., Svoboda, T., Van Gool, L., Lang, S., Strehlke, K., Vande Moere, A. and Staadt, O. (2003) *Blue-C: A Spatially Immersive Display and 3d Video Portal for Telepresence*, ACM SIGGRAPH 2003, San Diego, CA, California, pp.819-827.

Gross, M. H., Sprenger T. C. and J., F. (1997) *Visualizing Information on a Sphere*, IEEE Information Visualization '97, Phoenix, Arizona, pp.11-16.

Hansen, S. R., Narayanan, N. H. and Schrimpsher, D. (2000) *Helping Learners Visualize and Comprehend Algorithms*, Interactive Multimedia Electronic Journal of Computer-Enhanced Learning (1) 02.

Havre, S., Hetzler, E., Whitney, P. and Nowell, L. (2002) *Themeriver: Visualizing Thematic Changes in Large Document Collections*, Transactions on Visualization and Computer Graphics (8) 1, pp.9-20.

Healey, C. G. (1999) *Preattentive Processing in Visualization,* [Online], Available from: http://www.csc.ncsu.edu/faculty/healey/PP/PP.html, [Accessed: September, 2003].

Healey, C. G., Booth, K. S. and Enns, J. T. (1995) *Visualizing Real-Time Multivariate Data Using Preattentive Processing*, ACM Transactions on Modeling and Computer Simulation (ACM TOMACS) (5) 3, pp.190-221.

Healey, C. G. and Enns, J. T. (2002) *Perception and Painting: A Search for Effective, Engaging Visualizations*, IEEE Computer Graphics & Applications (22) 2, pp.10-15.

Heider, F. and Simmel, M. (1944) *An Experimental Study of Apparent Behavior*, American Journal of Psychology (57), pp.243–259.

Heim, M. (2000) *The Feng Shui of Virtual Reality*, Crossings: eJournal of Art and Technology (1) 1.

Hellerstein, J. M., Chou, A., Hidber, C., Olston, C., Raman, V., Roth, T. and Haas, P. J. (1999) *Interactive Data Analysis: The Control Project*, IEEE Computer (32) 8, pp.51.

Hin, A. J. S. and Post, F. H. (1993) *Visualization of Turbulent Flow with Particles*, IEEE Visualization, ACM, San Jose, California, pp.46 - 53.

Hinckley, K., Pausch, R., Goble, J. C. and Kassell, N. F. (1994) *A Survey of Design Issues in Spatial Input*, ACM symposium on User interface software and technology, ACM, Marina del Rey, California, United States, pp.213-222.

Hix, D. and Hartson, H. R. (1993) *Developing User Interfaces Ensuring Usability through Product & Process,* Wiley, New York etc.

Holmquist, L. E. and Skog, T. (2003) *Informative Art: Information Visualization in Everyday Environments*, Computer graphics and interactive techniques in Austalasia and South East Asia, ACM Press, Melbourne, Australia, pp.229-235.

Hultquist, J. P. M. (1992) *Constructing Stream Surfaces in Steady 3d Vector Fields*, IEEE Visualization (Vis'92), ACM Press, Boston, Massachusetts, pp.171-178.

Ilmonen, T. and Kontkanen, J. (2003) *The Second Order Particle System*, Journal of WSCG (11) 1.

Ishii, H. and Ullmer, B. (1997) *Tangible Bits: Towards Seamless Interfaces between People*, ACM CHI.

Jankun-Kelly, T. J., Ma, K.-L. and Gertz, M. (2002) *A Model for the Visualization Exploration Process*, IEEE Visualization, Boston, MA, USA.

Jaswal, V. (1997) *Cavevis: Distributed Real-Time Visualization of Time-Varying Scalar and Vector Fields Using the Cave Virtual Reality Theater*, Visualization '97, ACM Press, Phoenix, Arizona, United States, pp.301-ff.

Kadrovach, B. A. and Lamont, G. B. (2002) *A Particle Swarm Model for Swarm-Based Networked Sensor Systems*, ACM symposium on applied computing, ACM Press, Madrid, Spain, pp.918-924.

Kano, M., Tsutsumi, S., Nishimura, K., Aburatani, H., Hirota, K. and Hirose, M. (2002) *Visualization for Genome Function Analysis Using Immersive Projection Technology*, IEEE Virtual Reality Conference 2002, IEEE, Orlando, Florida.

Kenwright, D. N. and Lane, D. A. (1996) *Interactive Time-Dependent Particle Tracing Using Tetrahedral Decomposition*, IEEE Transactions on Visualization and Computer Graphics (2) 2, pp.120-129.

Kirner, T. G. and Martins, V. F. (2000) *Development of an Information Visualization Tool Using Virtual Reality*, Symposium on Applied Computing, ACM Press, Como, Italy, pp.604-606.

Knight, C. and Munro, M. (2000) *Virtual but Visible Software*, International Conference on Information Visualisation (IV2000), London, England, pp.198-205.

Kolers, P. A. (1972) *Aspects of Motion Perception,* Pergamon Press, Oxford.

Kramer, P. and Yantis, S. (1997) *Perceptual Grouping in Space and Time: Evidence from the Ternus Display*, Percept Psychophys (59) 1, pp.87-99.

Kuester, F., Bruckschen, R., Hamann, B. and Joy, K. I. (2001) *Visualization of Particle Traces in Virtual Environments*, ACM Symposium on Virtual Reality Software and Technology, ACM Press, Baniff, Alberta, Canada, pp.151- 157.

LAB_au (2003) *Lab[Au] Architecture and Urbanism, 0/1 Lab Metalab,* [Online], Available from: http://www.lab-au.com/files/doc/eu.htm, [Accessed: October, 2003].

Lamboray, E. (to appear) *A Communication Infrastructure for Highly-Immersive Collaborative Virtual Environments,* [Thesis], Department of Computer Science, ETH-Zurich.

Lamm, S. E., Reed, D. A. and Scullin, W. H. (1996) *Real-Time Geographic Visualization of World Wide Web Traffic*, Computer Networks and ISDN Systems (28) 7-11, pp.1457-1468.

Lander, J. (1998) *Ocean Spray in Your Face*, Game Developer, pp.9-13.

Lang, S. (to appear) *The Impact of Video Stream Systems on Architecture,* [Thesis], Department of Architecture, ETH-Zurich.

Lang, S., Naef, M., Gross, M. and Hovestadt, L. (2003) *In:Shop - Using Telepresence and Immersive Vr for a New Shopping Experience*, Vision, Modeling, Visualization (VMV'03), Munich, Germany.

Larkin, J. and Simon, H. (1987) *Why a Diagram Is (Sometimes) Worth Ten Thousand Words*, Cognitive Science (11), pp.65-99.

Laurel, B. (1991) *Computers as Theatre,* Addison-Wesley, Reading, Massachusetts etc.

Leigh, J. (1997) *Issues in the Design of a Flexible Distributed Architecture for Supporting Persistence and Interoperability in Collaborative Virtual Environments*, IEEE/ACM SC97 Conference, IEEE, San Jose, California, pp.21.

Leigh, J., Andrew, J., Brown, M., Sandin, D. J. and DeFanti, T. (1999a) *Visualization in Teleimmersive Environments*, IEEE Computer (32) 12, pp.66-73.

Leigh, J., Johnson, A., DeFanti, T., Bailey, S. and Grossman, R. (1999b) *A Tele-Immersive Environment for Collaborative Exploratory Analysis of Massive Data Sets*, ASCI 99, Heijen, The Netherlands, pp.3-9.

Leigh, J., Johnson, A., DeFanti, T. A. and Brown, M. (1999c) *A Review of Tele-Immersive Applications in the Cave Research Network*, IEEE Virtual Reality, Houston, Texas, pp.180-188.

Lethbridge, T. C. and Ware, C. (1990) Animation Using Behavior Functions, In *Visual Languages and Applications* (Eds, Ichikawa, T., Jungert, E. and Korfhage, R. R.), Plenum Press, New York, pp.237-252.

Levinson, B. (1994) *Disclosure*, Warner Bros., USA.

Levy, E., Zacks, J., Tversky, B. and Schiano, D. (1996) *Gratuitous Graphics? Putting Preferences in Perspective*, SIGCHI Conference on Human Factors and Computing Systems, Vancouver, British Columbia, Canada, pp.42 - 49.

Liang, J. and Green, M. (1994) *Jdcad: A Highly Interactive 3d Modeling System*, Computer & Graphics (4) 18, pp.499-506.

Lisberger, S. (1982) *Tron*, Buena Vista Pictures, USA.

Longo, R. (1995) *Johnny Mnemonic*, TriStar Pictures, USA.

Lorensen, W. E. and Cline, H. E. (1987) *Marching Cubes: A High Resolution 3d Surface Construction Algorithm*, International Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH 1987), ACM Press, pp.163-169.

Lum, E. B., Stompel, A. and Ma, K.-L. (2003) *Using Motion to Illustrate Static 3d Shape--Kinetic Visualization*, IEEE Transactions on Visualization and Computer Graphics (9) 2, pp.115-126.

Lyman, P. and Hal, R. V. (2003) *How Much Information,* [Online], Available from: http://www.sims.berkeley.edu/research/projects/how-much-info/, [Accessed: October, 2003].

Lynn, G. (1999) *Animate Form,* Princeton Architectural Press, New York.

Maas, W. and MvRdV (1999) *Metacity / Datatown,* 010 Publishers, Rotterdam.

Macgill, J. and Openshaw, S. (1998) *The Use of Flocks to Drive a Geographic Analysis Machine*, International Conference on GeoComputation, Bristol, United Kingdom.

Maeda, J. (2000) *Maeda @ Media,* Thames & Hudson, London.

Maletic, J. I., Marcus, A., Dunlap, G. and Jason, L. (2001) *Visualizing Object-Oriented Software in Virtual Reality*, International Workshop on Program Comprehension (IWPC'01), IEEE, pp.26-38.

Mander, R., Salomon, G. and Wong, Y. Y. (1992) *A "Pile" Metaphor for Supporting Casual Organization of Information*, Conference on Human Factors and Computing Systems (SIGCHI), ACM Press, Monterey, California, United States, pp.627-634.

Manovich, L. (2001a) *Information and Form,* [Online], Available from: http://www.manovich.net/ia/, [Accessed: September, 2003].

Manovich, L. (2001b) *The Language of New Media,* MIT Press, Cambridge, Mass.

Mark, G., Carpenter, K. and Kobsa, A. (2003) *A Model of Synchronous Collaborative Information Visualization*, International Conference on Information Visualization (IV'03), IEEE, London, England, pp.373-384.

Martin, A. (1999) *Particle Systems,* [Online], Available from: http://www.cs.wpi.edu/~matt/courses/cs563/talks/psys.html, [Accessed: August, 2003].

Mass, Y. and Herzberg, A. (1999) *Vrcommerce - Electronic Commerce in Virtual Reality*, ACM conference on Electronic commerce, ACM Press, Denver, Colorado, United States, pp.103-109.

McAllister, D. K. (2000) *The Design of an Api for Particle Systems*, (University of North Carolina at Chapel Hill, pp.8.

McAllister, D. K. (2001) *Particle System Api,* [Online], Available from: http://www.cs.unc.edu/~davemc/Particle/, [Accessed: October, 2003].

Merritt, R. K. (2000) *Memory Arts to the New Code Paradigm; the Artist Engineers Virtual Information Space and Virtual Experience*, 6th International Conference on Information Visualisation (IV02), IEEE Press, London, England, pp.239-243.

Merzkirch, W. (1987) *Flow Visualization,* Academic Press, New York.

Michotte, A. (1963) *The Perception of Causality,* Basic Books, New York.

Mieusset, K. H. (2000) *Supporting Team Awareness in a Collaboration*, Computer Aided Architectural Design Research in Asia (CAADRIA), Singapore, pp.25-33.

Mine, M. R. (2003) *Towards Virtual Reality for the Masses: 10 Years of Research at Disney's Vr Studio*, Workshop on Virtual Environments 2003, ACM Press, pp.11-17.

Mitchell, W. J. (1995) *City of Bits: Space, Place, and the Infobahn,* MIT Press, Massachusetts.

Moltenbrey, K. (1999) *Taking Stock*, Computer Graphics World (22) 10, pp.41-44.

Morrison, J. B., Tversky, B. and Bétrancourt, M. (2000) *Animation: Does It Facilitate Learning*, AAAI 2000 Spring Symposium Smart Graphics, Stanford, CA, USA, pp.53-60.

Mountford, J. (1992) Tools and Techniques for Creative Design, In *The Art of Human-Computer Interface Design* (Ed, Laurel, B.), Addison Wesley, Reading, Massachusetts etc, pp.17-30.

Müller, W. and Alexa, M. (1998) *Using Morphing for Information Visualization*, Workshop on New Paradigms in Information Visualization and Manipulation, ACM, pp.49-52.

Murray, J. H. (2000) *Hamlet on the Holodeck: The Future of Narrative in Cyberspace,* MIT Press, Cambridge.

Naef, M. (to appear) *Blue-C Api: Software Support for Collaborative Virtual Reality,* [Thesis], Department of Computer Science, ETH-Zurich.

Naef, M., Lamboray, E., Staadt, O. and Gross, M. (2003) *The Blue-C Distributed Scene Graph*, IPT/EGVE 2003, ACM Press, Zurich, Switzerland, pp.125-133.

Nagel, H. R. and Granum, E. (2002) *A Software System for Temporal Data Visualization in Virtual Reality*, Software System for Temporal Data Visualization in Virtual Reality, Workshop on Data Visualization for large data sets and Data Mining, Augsburg, Germany.

Nakayama, K. and Silverman, G. H. (1986) *Serial and Parallel Processing of Visual Feature Conjunctions*, Nature (320) 6059, pp.264-265.

Negroponte, N. (1996) *Being Digital,* Knopf, New York.

Nielsen, J. (1993) *Usability Engineering,* AP Professional, Boston, MA.

NIST/SEMATECH (2003) *E-Handbook of Statistical Methods,* [Online], Available from: http://www.itl.nist.gov/div898/handbook, [Accessed: August, 2003].

Norman, D. A. (1993) *Things That Make Us Smart,* Addison-Wesley, Reading, Massachussets.

Norman, D. A. (2003) *Attractive Things Work Better,* [Online], Available from: http://www.jnd.org/ED_Draft/CH01.pdf, [Accessed: October, 2003].

Novak, M. (2003) *Centrifuge,* [Online], Available from: http://www.centrifuge.org/marcos/, [Accessed: October, 2003].

Palter, A. M., Stone, P. A., Meier, B. J., Miller, T. S. and Simpson, R. M. (2000) *Interaction in an Ivr Museum of Color*, SIGGRAPH 2000 - Abstracts and Applications, ACM Press, pp.41–44.

Pang, A. (1994) *Spray Rendering*, IEEE Computer Graphics & Applications (14) 5, pp.57-63.

Pang, A. and Smith, K. (1993) *Spray Rendering: Visualization Using Smart Particles*, IEEE Visualization, IEEE, San Jose, California, pp.283-290.

Parent, R. (1998) *Algorithms and Models to Fully Specify Motion,* [Online], Available from: http://www.cis.ohio-state.edu/~parent/book/Modl.html, [Accessed: August, 2003].

Pausch, R., Proffitt, D. and Williams, G. (1997) *Quantifying Immersion in Virtual Reality*, International Conference on Computer Graphics and Interactive Techniques, ACM, pp.13-18.

Pettifer, S., Cook, J. and Mariani, J. (2001) *Towards Real-Time Interactive Visualisation in Virtual Environments: A Case Study of Q-Space*, International Conference on Virtual Reality 2001, Laval.

Pettifer, S. and West, A. (1997) *Deva: A Coherent Operating Environment for Large Scale Vr Applications*, Virtual Reality Universe, California, Santa Clara.

Plaisant, C., Milash, B., Rose, A., Widoff, S. and Shneiderman, B. (1996) *Lifelines: Visualizing Personal Histories*, Conference on Human Factors in Computing Systems (CHI'96), ACM Press, Vancouver, BC, Canada, pp.221-227.

Proctor, G. and Winter, C. (1998) *Information Flocking: Data Visualisation in Virtual Worlds Using Emergent Behaviours*, Virtual Worlds 98, Springer, Paris, France, pp.168-176.

Proffit, D. B. and Kaiser, M. K. (1995) Perceiving Events, In *Perception of Space and Motion* (Eds, Epstein, W. and Rogers, S.), Academic Press, London, pp.227-261.

Redström, J., Skog, T. and Hallnäs, L. (2000) *Informative Art: Using Amplified Artworks as Information Displays*, Designing Augmented Reality Environments (DARE 2000), ACM Press, Elsinore, Denmark, pp.103-114.

Reed, D. A., Giles, R. C. and E., C. C. (1997) *Distributed Data and Immersive Collaboration*, Communications of the ACM (40) 11, pp.38-48.

Reeves, W. T. (1983) *Particle Systems-a Technique for Modeling a Class of Fuzzy Objects*, Computer-Graphics (17) 3, pp.359-76.

Reeves, W. T. and Blau, R. (1985) *Approximate and Probabilistic Algorithms for Shading and Rendering Structured Particle System*, Computer Graphics (19) 3, pp.313-322.

Reynolds, C. W. (1987) *Flocks, Herds, and Schools: A Distributed Behavioral Model*, Computer Graphics (21) 4, pp.25-34.

Reynolds, C. W. (2001) *Boids (Flocks, Herds, and Schools: A Distributed Behavorial Model),* [Online], Available from: http://www.red3d.com/cwr/boids/, [Accessed: October, 2003].

Ribarsky, W., Katz, J., Jiang, F. and Holland, A. (1999) *Discovery Visualization Using Fast Clustering*, IEEE Computer Graphics & Applications (19) 5, pp.32-39.

Robertson, G. G., Card, S. K. and Mackinlay, J. D. (1993) *Information Visualization Using 3d Interactive Animation*, Communications of the ACM (36) 4, pp.57-71.

Robertson, G. G., Czerwinski, M., Larson, K., Robbins, D., Thiel, D. and van Dantzich, M. (1998) *Data Mountain: Using Spatial Memory for Document Management*, ACM Symposium on User Interface Software and Technology (UIST), ACM Press, San Francisco, CA, pp.153-162.

Rohlf, J. and Helman, J. (1994) *Iris Performer: A High Performance Multiprocessing Toolkit for Real-Time 3d Graphics.*, ACM SIGGRAPH, pp.381-395.

Rohrer, R. M. and Sibert, J. L. (1998) *The Shape of Shakespeare: Visualizing Text Using Implicit Surfaces*, IEEE Symposium on Information Visualization, North Carolina, pp.121-130.

Rojdestvenski, I., Modjeska, D., Pettersson, F., Rojdestvenskaia, M. and Gustafsson, P. (2000) *Sequence World: A Genetics Database in Virtual Reality*, International Conference on Information Visualisation (IV2000), IEEE, London, England, pp.513-517.

Rosenblum, L. J. and Post, F. H. (1993) *Visualizing Fluid Flow*, Computer (26) 6, pp.98-100.

Rossi, A. M. and Varga, M. (1999) *Visualization of Massive Retrieved Newsfeeds in Interactive 3d*, International Conference on Information Visualisation (IV1999), London, England, pp.12-17.

Russo Dos Santos, C., Gross, P., Abel, P., Loisel, D., Trichaud, N. and Paris, J. P. (2000) *Mapping Information onto 3d Virtual Worlds*, International Conference on Information Visualisation (IV2000), London, England, pp.379-388.

Sadagic, A., Towles, H., Lanier, J., Fuchs, H., van Dam, A., Daniilidis, K., Mulligan, J., Holden, L. and Zeleznik, B. (2001) *National Tele-Immersion Initiative: Towards Compelling Tele-Immersive Collaborative Environments*, Medicine meets Virtual Reality 2001 Conference, Newport Beach, California.

Sawant, N., Scharver, C., Leigh, J., Johnson, A., Reinhart, G., Creel, E., Batchu, S., Bailey, S. and Grossman, R. (2000) *The Tele-Immersive Data Explorer: A Distributed Architecture for Collaborative Interactive Visualization of Large Data-Sets*, International Immersive Projection Technology Workshop, Ames, Iowa.

Scanlon, J. (1999) *Ride the Dow*, Wired 7.06.

Schmitt, G. N. (1999) *Information Architecture Basis and Future of Caad,* Birkhäuser, Basel.

Schmitt, G. N., Wenz, F., Kurmann, D. and van der Mark, E. (1995) *Towards Virtual Reality in Architecture: Concepts and Scenarios from the Architectural Space Laboratory*, Presence Journal, Massachusetts of Technology (4) 3, pp.267-285.

Schroeder, W. J., Volpe, C. R. and Lorensen, W. E. (1991) *The Stream Polygon: A Technique for 3d Vector Field Visualization*, IEEE Visualization 1991, pp.126-132.

science_museum (1997) *Electrons in Atoms,* [Online], Available from: http://www.sciencemuseum.org.uk/on-line/electron/section3/1937.asp, [Accessed: October, 2003].

Sebrechts, M. M., Vasilakis, J., Miller, M. S., Cugini, J. V. and Laskowski, S. J. (1999) *Visualization of Search Results: A Comparative Evaluation of Text, 2d, and 3d Interfaces*, Conference on Research and Development in Information Retrieval (ACM SIGIR), Berkeley, California, United States.

Shaw, C. D., Ebert, D. S., Kukla, J. M., Zwa, A., Soboroff, I. and Roberts, D. A. (1998) *Data Visualization Using Automatic, Perceptually-Motivated Shapes*, Visual Data Exploration and Analysis V, The Society of Imaging Science & Technology, SPIE - The International Society for Optical Engineering, San Jose, California.

Shaw, C. D., Kukla, J. M., Soboro, I., Ebert, D. S., Nicholas, C. K., Zwa, A., Mille, E. L. and Roberts, D. A. (1999) *Interactive Volumetric Information Visualization for Document Corpus Management*, International Journal on Digital Libraries (2), pp.144-156.

Sherman, B. and Judkins, P. (1992) *Glimpses of Heaven, Visions of Hell - Virtual Reality and Its Implications,* Hodder and Stoughton.

Sherman, W. R. and Craig, A. B. (2003) *Understanding Virtual Reality - Interface, Application, and Design,* Morgan Kaufmann Publishers, Amsterdam.

Shneiderman, B. (1994) *Dynamic Queries for Visual Information Seeking*, IEEE Software (11) 6, pp.70-77.

Shneiderman, B. (1996) *The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations*, 1996 IEEE, Visual Languages, Boulder, CO, pp.336-343.

Shneiderman, B. (1998) *Designing the User Interface: Strategies for Effective Human-Computer Interaction,* Addison-Wesley Publishing.

Shneiderman, B. (2003) *Why Not Make Interfaces Better Than 3d Reality?*, IEEE Computer Graphics and Applications (23) 6, pp.12-15.

Sifer, M. (2002) *Exploring Web Site Log Data with a Multi-Classification Interface*, International Conference on Information Visualization (IV'03), IEEE, London, England, pp.94-102.

Sims, K. (1990) *Particle Animation and Rendering Using Data Parallel Computation*, International Conference on Computer Graphics and Interactive Techniques, ACM Press, Dallas, TX, USA, pp.405-413.

Skelly, T., Fries, K., Linnett, B., Nass, C. and Reeves, B. (1994) *Seductive Interfaces: Satisfying a Mass Audience*, Conference on Human Factors and Computing Systems, ACM Press, Boston, Massachusetts, United States, pp.359-360.

Small, D. (1998) *Rethinking the Book,* [Thesis], School of Architecture and Planning, Massachusetts Institute of Technology (MIT).

Smith, K. U., Smith, W. M. and Smith, M. F. (1962) *Perception and Motion an Analysis of Space-Structured Behavior,* Saunders, Philadelphia - Pa. London.

Song, D. and Norman, M. L. (1993) *Cosmic Explorer: A Virtual Reality Environment for Exploring Cosmic Data*, IEEE Symposium on Research Frontiers in Virtual Reality, IEEE, San Jose, CA.

Spagno, C. (2003) *Immersive Vr Projection System with Simultaneous Image Acquisition Using Active Projection Screens,* [Thesis], Product Development, ETH-Zurich.

Sparacino, F., DeVaul, R., Wren, C., MacNeil, R., Davenport, G. and Pentland, A. (1999) *City of News*, ACM SIGGRAPH (Visual Proceedings, Emerging Technologies), Los Angeles, USA.

Sprenger, T. C. (2002) *Ivory - an Object-Oriented Framework for Physics-Based Information Visualization,* [Thesis], Departement of Computer Science, ETH-Zurich.

Sprenger, T. C., Gross, M., Eggenberger, A. and Strasser, T. (1998) *Ivory - an Object-Oriented Framework for Physically-Based Information Visualization in Java*, IEEE Information Visualization (INFOVIS'98), IEEE, NC, USA, pp.79-86.

Stasko, J. T. (1993) Animation in User Interfaces: Principles and Techniques, In *Trends in Software, Special Issue on User Interface Software*, Vol. 1 (Eds, Bass, L. and Dewan, P.), pp.81-101.

Stouffs, R., Kurmann, D., Tuncer, B., Mieusset, K. H. and Staeger, B. (1998) *An Information Architecture for the Virtual Aec Company*, Product and Process Modelling in the Building Industry, Watford, UK, pp.479-486.

Sutherland, I. E. (1968) *A Head-Mounted Three-Dimensional Display*, AFIPS Conference, pp.757-764.

SWCP (2003) *Historical Data for S&P 500 Stocks,* [Online], Available from: http://kumo.swcp.com/stocks, [Accessed: October, 2003].

Takada, T. and Koike, H. (2002) *Tudumi: Information Visualization System for Monitoring and Auditing Computer Logs*, International Conference on Information Visualisation (IV'02), IEEE, London, England, pp.570-577.

Taskaya, T. and Ahmad, K. (2003) *Bimodal Visualisation: A Financial Trading Case Study*, IEEE International Conference on Information Visualization (IV'03), IEEE, London, England, pp.320-327.

Tonnesen, D. (2001) *Particle Systems for Artistic Expression*, Subtle Technologies Conference, Toronto, Canada, pp.17-20.

Tonnesen, D. L. (1998) *Dynamically Coupled Particle Systems for Geometric Modeling, Reconstruction and Animation,* [Thesis], Department of Computer Science, University of Toronto.

Tversky, B., Morrison, J. B. and Betrancourt, M. (2002) *Animation: Can It Facilitate?*, International Journal of Human-Computer Studies (57) 4, pp.247-262.

UCAR (2000) *The Trajectory of Comet Shoemaker-Levy 9 over Time,* [Online], Available from: http://www.windows.ucar.edu/tour/link=/comets/SL9_trajectory.html, [Accessed: October, 2003].

unified_field (2003) *4d-Visual (Unified Field Division),* [Online], Available from: http://www.unifiedfield.com/4d/, [Accessed: October, 2003].

Upson, C., Faulhaber, T. A., Jr., Kamins, D., Laidlaw, D., Schlegel, D., Vroom, J., Gurwitz, R. and van Dam, A. (1989) *The Application Visualization System: A Computational Environment for Scientific Visualization*, (9), pp.30-42.

V2 (2003) *V2_Organization, Institute for Unstable Media,* [Online], Available from: http://www.v2.nl/, [Accessed: October, 2003].

van Berkel, B. and Bos, C. (1999) *Move - Un Studio,* Architectura & Natura.

van Dam, A., Forsberg, A. S., Laidlaw, D. H., LaViola, J. J. and Simpson, R. M. (2000) *Immersive Vr for Scientific Visualization: A Progress Report*, IEEE Computer Graphics and Applications (20) 6, pp.26-52.

van der Burg, J. (2000) *Building an Advanced Particle System*, Game Developer, pp.44-50.

Vande Moere, A. (1998) *The Language of Cyberspace: An Architectural Approach,* [Thesis], Department of Architecture, Urban Design and Regional Planning, K.U.Leuven.

Vande Moere, A. (2001) 3.D.H.T.M.L., In *Bits and Spaces - Architecture and Computing for Physical, Virtual, Hybrid Realms - 33 Projects by Architecture and Caad, Eth Zurich* (Ed, Engeli, M.), Birkhäuser, Basel.

Vaughan, L. C. (1997) *Understanding Movement*, ACM SIGCHI 1997, ACM Press, pp.548-549.

Wachowski, A. and Wachowski, L. (1999) *The Matrix*, Warner Bros., USA.

Wakita, A. and Matsumoto, F. (2003) *Information Visualization with Web3d - Spatial Visualization of Human Activity Area and Its Condition*, SIGGRAPH Computer Graphics (37) 3, pp.29-33.

Ware, C. (2000) *Information Visualization Perception for Design,* Morgan Kaufmann.

Ware, C. and Franck, G. (1996) *Evaluating Stereo and Motion Cues for Visualizing Information Nets in Three Dimensions*, ACM Transactions on Graphics, pp.121-140.

Ware, C., Neufeld, E. and Bartram, L. (1999) *Visualizing Causal Relations*, IEEE Symposium on Information Visualization (Infovis'99), San Francisco, CA, pp.39-42.

Wattenberg, M. (2002) *Arc Diagrams: Visualizing Structure in Strings*, IEEE Symposium on Information Visualization (Infovis'02), IEEE, Boston, MA, pp.110-116.

Wei, X., Kaufman, A. and Hallman, T. J. (2001) *Case Study: Visualization of Particle Track Data*, IEEE Visualization 2001 Conference (VIS 2001), IEEE, San Diego, CA, pp.??

Wertheimer, M. (1923) *Untersuchungen Zur Lehre Von Der Gestalt*, Psychologische Forschung (4), pp.301-350.

Westerlund, B. (2002) *Form Is Function*, Designing Interactive Systems (DIS2002), London, England, pp.117-124.

Wills, G. J. (1996) *524,288 Ways to Say "This Is Interesting"*, IEEE Symposium on Information Visualization (Infovis'96), pp.54-60.

Wise, J. A., Thomas, J. J., Pennock, K., Lantrip, D., Pottier, M., Schur, A. and Crow, V. (1995) *Visualizing the Non-Visual: Spatial Analysis and Interaction with Information from Text Documents*, IEEE Information Visualization, pp.51-58.

Wisneski, C., Ishii, H., Dahley, A., Gorbet, M., Brave, S., Ullmer, B. and Yarin, P. (1998) *Ambient Displays: Turning Architectural Space into an Interface between People and Digital Information.*, First International Workshop on Cooperative Buildings (CoBuild '98), Springer-Verlag, pp.22-32.

Witkin, A. P. and Heckbert, P. S. (1994) *Using Particles to Sample and Control Implicit Surfaces*, International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'94), ACM Press, pp.269-277.

Wurman, S. R. (2000) *Understanding USA,* [Online], Available from: http://www.understandingusa.com, [Accessed: October, 2003].

Würmlin, S. (to appear) *Dynamic Point Samples as Primitives for Free-Viewpoint Video,* [Thesis], Department of Computer Science, ETH-Zurich.

Young, P. and Munro, M. (1998) *Visualising Software in Virtual Reality*, 6th IWPC, Ischia, Italy, pp.17-24.

E. Curriculum

Personal Details

Name	Vande Moere, Andrew
Nationality	Belgium
Email	andrew@ticle.com

Education

2000 - 11/2003	Ph.D. Student ETH-Zurich, Switzerland.
1998 – 1999	Post-Graduate Degree in Computer Aided Architectural Design Chair for CAAD, ETH-Zurich, Switzerland.
1993 - 1998	Master in Architectural Engineering K.U.Leuven University, Belgium. thesis title: “The Language of Cyberspace: an Architectural Approach”.
1987 - 1993	Mathematics (8h) / Sciences K.A.Boom, Belgium.

Work Experience

2004 - …	Lecturer in Design Computing, University of Sydney.
1999 - 2004	Research Assistant at the blue-c project, ETH-Zurich.
11-12/2002	Visiting Researcher at ARUP Research & Development, London.
1998-2001	Teaching & Research Assistant, Chair of CAAD & Architecture, ETH-Zurich.

Languages

English	fluent speaking + writing + reading
German	fluent speaking + reading, moderate writing
French	moderate speaking + writing + reading
Dutch	mother tongue

Peer-Reviewed Publications

2004 Andrew Vande Moere, *Time-Varying Data Visualization using Information Flocking Boids*, in proc. of IEEE Symposium on Information Visualization (INFOVIS'04), Austin, Texas, October 2004, accepted for publication.

2004 Andrew Vande Moere, *Information Flocking: Time-Varying Data Visualization using Boid Behaviors*, in proc. of the 8th International Conference on Information Visualization (IV'04), London, July 2004, accepted for publication.

2004 Andrew Vande Moere, Kuk Hwan Mieusset, *Visualizing Abstract Information using Motion Properties of Data-Driven Infoticles*, Conference on Visualization and Data Analysis, IS&T/SPIE Symposium on Electronic Imaging 2004, San Jose, January 2004.

2003 Andrew Vande Moere, Kuk Hwan Mieusset, *Translating Data Updates Into Dynamic Particle Behavior*, HCI2003 Workshop on Metaphor and HCI, Bath, Sept. 2003, accepted for publication.

2003 Markus Gross, Stephan Würmlin, Martin Naef, Edouard Lamboray, Christian Spagno, Andreas Kunz, Esther Koller-Meier, Tomas Svoboda, Luc Van Gool, Silke Lang, Kai Strehlke, Andrew Vande Moere, Oliver Staadt, *blue-c: A Spatially Immersive Display and 3D Video Portal for Telepresence*, in proc. of ACM SIGGRAPH 2003, San Diego, July 2003, pp.819-827.

2002 Andrew Vande Moere, *Interactive Poster: Immersive Information Modeling using Particles*, Poster Comp. of IEEE Symposium on Information Visualization (INFOVIS'02), Boston, Oct. 2002, pp.32-33.

2002 Andrew Vande Moere, *Infoticles: Information Modeling in Immersive Environments*, in proc. of the 6th International Conference on Information Visualization (IV'02), London, July 2002, pp.457-461.

2001 Andrew Vande Moere, Patrick Keller, *Recombinant Realities: the Design of a Multi-User Environment in Cyberspace*, in proc. of CAST01 Living in Mixed Realities, Bonn, Sept. 2001, pp.383-386.

2001 Mishka Bugajska, Andrew Vande Moere, *Virtual Library: Paths to Knowledge*, in proc. of 2nd Int. Workshop on User Interfaces to Data Intensive Systems (UIDIS'01), Zurich, May 2001, pp.116-126.

1998 Andrew Vande Moere, Herman Neuckermans, Ann Heylighen, *VR/Search: An Architectural Approach to Cyberspace*, in C. Branki, G. Vasquez de Velasco, International Journal of Design Sciences and Technology, Vol.7 - N°1, 1998, pp.23-39.

1998 Andrew Vande Moere, Herman Neuckermans, Ann Heylighen, *VR/Search: An Architectural Approach to Cyberspace*, in C. Branki & K. Zreik, Cyber Design, EuropIA'98, Nov. 1998, Paris, pp.223-229.

Other Publications

2003 Andrew Vande Moere, *Infoticles*, Interface Theory. anomalie_3, Paris, May 2003.

2001 Andrew Vande Moere, *3.D.H.T.M.L.,* in bits and spaces: Architecture and Computing for the Virtual, Physical, Hybrid Realm - 33 Projects by Architecture and CAAD ETH Zurich, editor: Maia Engeli, Basel, 2001, Birkhäuser Publishers.

2001 Andrew Vande Moere, *Recombinant Realities,* in bits and spaces: Architecture and Computing for the Virtual, Physical, Hybrid Realm - 33 Projects by Architecture and CAAD ETH Zurich, editor: Maia Engeli, Basel, 2001, Birkhäuser Publishers.

2001 Andrew Vande Moere, *Virtual Library*, in bits and spaces: Architecture and Computing for the Virtual, Physical, Hybrid Realm - 33 Projects by Architecture and CAAD ETH Zurich, editor: Maia Engeli, Basel, 2001, Birkhäuser Publishers.

1999 Andrew Vande Moere, *3.D.N.A.: a 3D Visualization and Interaction Application of abstract HTML Data Structures*, Post-Graduate Thesis, ETH-Zurich, June 1999.

1998 Andrew Vande Moere, *The Language of Cyberspace: An Architectural Approach*, Master Thesis, K.U.Leuven University, Belgium, June 1998.