Interaction Design for Complex Cognitive Activities with Visual Representations: A Pattern-Based Approach

Kamran Sedig  
Western University  
sedig@uwo.ca

Paul Parsons  
Western University  
pparsons@uwo.ca

Abstract

This paper is concerned with interaction design for visualization-based computational tools that support the performance of complex cognitive activities, such as analytical reasoning, sense making, decision making, problem solving, learning, planning, and knowledge discovery. In this paper, a number of foundational concepts related to interaction and complex cognitive activities are syncretized into a coherent theoretical framework. This framework is general, in the sense that it is applicable to all technologies, platforms, tools, users, activities, and visual representations. Included in the framework is a catalog of 32 fundamental epistemic action patterns, with each action pattern being characterized and examined in terms of its utility in supporting different complex cognitive activities. This catalog of action patterns is comprehensive, covering a broad range of interactions that are performed by a diverse group of users for all kinds of tasks and activities. The presented framework is also generative, in that it can stimulate creativity and innovation in research and design for a number of domains and disciplines, including data and information visualization, visual analytics, digital libraries, health informatics, learning sciences and technologies, personal information management, decision support, information systems, and knowledge management.

Keywords: Interactive visualizations, design framework, tasks and activities, cognitive activity support tools, action taxonomy, epistemic actions, human-information interaction, complex cognition

Ping Zhang was the accepting Senior Editor. This article was submitted on 3/5/2012 and accepted on 1/5/2013. It was with the authors 189 days for 2 revisions.

INTRODUCTION

Many common activities nowadays are information-intensive and involve complex human cognition (Funke, 2010; Sternberg and Ben-Zeév, 2001). Such activities include decision making, problem solving, sense making, learning, and analysing, all of which can be referred to as complex cognitive activities (see Ericsson and Hastie, 1994). Knauff and Wolf (2010) identified two essential characteristics of complex cognitive activities: 1) the use of complex psychological processes, and 2) the presence of complex conditions. That is, complex cognitive activities are emergent and rely on the combination and interaction of more elementary processes such as perception and memory. Furthermore, they may involve many variables that exhibit a high level of interdependence and may change over time.

In recent years, computational tools and technologies have become deeply embedded in the performance of many complex cognitive activities (Dascal and Dror, 2005). These technologies play two important roles: epistemic and ontic (Brey, 2005). In their epistemic role, they act as tools that extend, partner, supplement, and support human cognitive faculties and functioning by maintaining, operating upon, and displaying digital information. In their ontic role, they act as tools that extend the world and simulate it. In this paper, we are concerned with the epistemic role of these computational tools, particularly as it relates to the aforementioned complex cognitive activities. In this role, depending on the context for which they are used, different terms are used to refer to them, such as cognitive technologies, decision support systems, knowledge support systems, cognitive tools, learning support tools, mind tools, and the like (e.g., Markus et al., 2002; Fischer and Sharff, 1998; Kim and Reeves, 2007; Bhargava et al., 2007; Sedig and Liang, 2008). As the focus here is on tools that mediate and enable complex cognitive activities, we unify all of them under one umbrella term to emphasize their epistemic support role, and will henceforth refer to all such tools as cognitive activity support tools (CASTs). In this context, the term ‘support’ suggests that CASTs can partner, distribute, augment, amplify, channelize, guide, offload, cognize with, shape, and/or transform human activities and thinking. Additionally, we broadly refer to people who use these tools as users, even though other terms, such as knowledge worker, learner, problem solver, analyzer, planner, or decision maker, can be contextually more accurate.

To perform complex cognitive activities, different types of CASTs with varying degrees of complexity are used. Such tools include library and research collection tools, drug analysis tools, knowledge mapping tools, financial analysis tools, virtual science museums, genome analysis tools, mathematical investigation software, social network visualizations, geovisualization tools, crime analysis tools, public health informatics tools, and business modelling tools (see e.g., MacEachren et al., 2004; Fast and Sedig, 2011; Shannon et al., 2003; Wagner, 2003; Xu and Chen, 2005; Thomas and Cook, 2005; Sedig et al., 2012b). The information with which users of CASTs interact can originate from all kinds of concrete or abstract sources, such as genes and other biological phenomena, historical records, scientific experiments, mathematical objects and processes, hospitals and medical clinics, research and social networks, library collections, and financial markets. For the purposes of this paper, the common feature of all CASTs is that they have a visually perceptible interface that mediates between a user and information. This is done by providing interaction mechanisms through which users can access and process displayed representations of information as well as input new information into the CASTs. Hence, these tools participate in the interplay between external representations of information and internal mental representations of users. Accordingly, the epistemic locus of CASTs is their information interface (Lovett and Shah, 2007; Sedig, 2008).

Viewed from this perspective, we can identify two main components that comprise the information interface of CASTs: representation and interaction (Sedig, 2004; Yi et al., 2007). Design of the representation component of CASTs is concerned with how information can and should be encoded and displayed. The purpose of this component is to support users in their perceptual as well as their cognitive processing of the information. Design of the interaction component of CASTs is concerned with what users can and should do with the represented information, what actions should be made available to them to work and think with the represented information, and what their subsequent reactions should be. The focus of interaction design, then, is on the discourse that takes place between users and the represented information. It is through interaction with the represented information that users can restructure and modify the form and amount of displayed information in order to optimize and enhance its epistemic utility for performing complex cognitive activities. Hence, the representation and interaction components of the interface are at the heart of the epistemic role that CASTs play. The proper and systematic design of both these components, then, determines the degree of epistemic utility of CASTs, and how well they support their users’ cognitive processes and activities (Sedig et al., 2001, 2003, 2005; Liang and Sedig, 2010; Thomas and Cook, 2005).

Although some fundamental concepts and techniques regarding representation and interaction design have been in place for a while (e.g., Bertin, 1983; Beynon et al., 2001; Lohse et al., 1994; Shneiderman, 1991; Tufte, 1983; Yi et al., 2007), many researchers suggest that we do not yet have a generalized, principled, and systematic understanding of the representation and interaction components of CASTs, and how these two components should be analyzed, designed, and integrated to support complex cognitive activities. For instance, in their seminal report on visual analytics, Thomas and Cook (2005) stated that “we lack fundamental understanding of the basic principles for
effectively conveying information using graphical techniques” (p. 70), and that “although a lot of isolated design work has been done in specific aspects of interaction science, little systematic examination of the design space has been done” (p. 76). While this was stated a number of years ago, numerous researchers have more recently suggested that this is an extant issue. Consider the following statements:

- “the process of stimulating and enabling human reasoning with the aid of interactive visualization tools is still a highly unexplored field.” (Meyer et al., 2010, p. 227);
- “with all of this research, there is still a lack of precedent on how to conduct research into visually enabled reasoning. It is not at all clear how one might evaluate interfaces with respect to their ability to scaffold higher-order cognitive tasks.” (Green and Fisher, 2011, p. 45);
- “we still know little about the effectiveness of graphic displays for space-time problem solving and behavior, exploratory data analysis, knowledge exploration, learning, and decision-making”. (Fabrikant, 2011, p. 1);
- “there is hardly ever an explanation of what these benefits [of interaction] actually are as well as how and why they work.” (Aigner, 2011, p. 18); and,
- “we have barely scratched the surface of this exciting new line of research [regarding interaction], and much work remains to be done.” (Elmqvist et al., 2011, p. 337).

What becomes evident from surveying existing literature is that research that does exist is insufficient, and that there are no comprehensive frameworks that support researchers and practitioners in terms of understanding how these two components relate in the context of performing complex cognitive activities. The relevant body of research dealing with these components is fragmentary and scattered across a set of disciplines (such as cognitive and learning sciences, information visualization, educational technologies, visual analytics, and human-computer interaction), often involving minimal interaction and collaboration between them.

Many researchers concerned with different facets of human-information interaction have recently suggested that a necessary theoretical substrate is not well developed (e.g., Fidel, 2012; Kapitelin and Nardi, 2012; Liu and Stasko, 2010; Purchase et al., 2008; Sedig et al., 2013). Such is to be expected, however, as this is a relatively young research area. Indeed, it is typically the case that the theoretical scope of any scientific discipline expands to become generally applicable and to encompass a wide range of phenomena only after an initial phase of specialization and division (Bohm and Peat, 1987). While discussing the development of scientific theories, von Baeyer suggests that “increased abstraction is the hallmark of growing maturity” (2004, p. 36) of any scientific discipline. Recent emphasis on the need for a more coherent and abstract theory of interaction and its related areas may signify an important stage in the evolution and growing maturity of human-information interaction research. The development of such a theoretical framework is one of the goals of this paper.

Throughout the past decade, a number of researchers have been working on the development of frameworks dealing with different aspects and levels of interaction design, such as benefits, costs, activities, techniques, and tasks (Amar et al., 2005; Amar and Stasko, 2005; Gotz and Zhou, 2008; Lam, 2008; Sedig and Sumner, 2006; Shrinivasan and van Vijk, 2008; Liu and Stasko, 2010; Nakakoji and Yamamoto, 2003; Yi et al., 2007; Pike et al., 2009; Fast and Sedig, 2011). Much of this research has been in the context of specific domains and tools, such as visual analytics and information visualization (e.g., Pike et al., 2009; Gotz and Zhou, 2008; Liu and Stasko, 2010). Being focused only on certain types of CASTs and domains, this research has analyzed a limited set of complex cognitive activities (e.g., analytical reasoning and sense making) and tasks (e.g., computing derived values, determining range, and finding extreme values). As valuable as this research is, it has not been—and cannot be—generalized beyond these CASTs and their pertinent activities, tasks, representations, interactions, and users, to sufficiently address the theoretical need discussed above. For instance, an oft-quoted and valuable prescription “overview first, zoom and filter, details-on-demand” (Shneiderman, 1996), which is applicable to some information visualization tools, does not necessarily generalize to other types of CASTs and their related representations and complex cognitive activities (e.g., see Sedig et al., 2001).

This paper presents a framework that supports systematic thinking about interaction design for complex cognitive activities. The framework has been developed to have the following four important characteristics, which position it to address existing research gaps and challenges and to make a valuable contribution to the existing literature: 1) to be syncretic, unifying a number of previously disconnected ideas into a coherent theoretical model; 2) to be general, operating at a level of abstraction that is applicable to all kinds of technologies, activities, users, and visual representations; 3) to be comprehensive, identifying patterns that cover an extensive range of actions; and, 4) to be generative, possessing the ability to motivate design creativity as well as to stimulate further theoretical and applied research.

No one framework or paradigm can address all possible tasks and situations (Purchase et al., 2008; Thomas and Cook, 2005). Accordingly, this paper is not complete in its characterization, but is a part of a broader research agenda to develop a comprehensive, principled, and systematic framework concerned with the information interface of CASTs, called EDIFICE (Epistemology and Design of human-InFormation Interaction in complex Cognitive activities). This paper complements the other components of EDIFICE, which include: 1) a framework dealing with...
the design of visual representations, 2) a framework dealing with the analysis of the ontological properties of visual representations that affect the performance of complex cognitive activities, and 3) a framework dealing with the detailed analysis of the anatomical structure of an individual interaction as well as the manner in which interactions are combined and integrated during the performance of complex cognitive activities. The component of EDIFICE that is developed in this paper deals with the interaction design of CASTs. The interaction that takes place between a user and a CAST can be characterized at multiple levels of granularity (see Sedig et al., 2013). Such levels include macro-level activities, tasks, individual actions and reactions (i.e., interactions), and micro-level events. Even though this paper discusses interaction at all these levels, its focus is mainly on interaction at the level of individual actions and reactions, dealing primarily with pattern-based characterizations of actions and their utility in supporting complex cognitive activities. Because this framework is human-centered, focusing on the action component of interaction, and because it takes a pattern-based approach, we will henceforth refer to it as EDIFICE-AP (where AP stands for Action Patterns).

The rest of this paper is divided into 3 main sections: 1) conceptual and theoretical foundations; 2) the EDIFICE-AP framework; and 3) summary and future directions.

CONCEPTUAL AND THEORETICAL FOUNDATIONS

This section serves a twofold function. First, it examines the utility of, and need for, frameworks; second, it identifies and explicates a number of terms and concepts that are necessary for discussing human-information interaction in the context of CASTs. These terms and concepts have been used in different contexts often with different meanings and connotations. It is necessary, therefore, to characterize them and examine their relationships before presenting the EDIFICE-AP framework.

Frameworks

Since the time of Plato and Aristotle, frameworks and classification systems have played an important role in systematic and scientific exploration of phenomena (Darian, 2003). Conducting research to develop frameworks, taxonomies, and models is crucial to the advancement of any discipline, including the analysis and design of computational tools (Carroll, 1991; Heller et al. 2001; Hult et al., 2006). Bederson and Shneiderman (2003) enumerate five roles for this type of research: 1) describe (characterize objects and actions in a systematic manner to provide clear language and enable cooperation), 2) explain (explain processes to support education), 3) predict (predict performance in different situations), 4) prescribe (suggest guidelines and best practices), and 5) generate (facilitate innovation). In the case of interaction design for CASTs, a framework concerned with epistemic action patterns can serve each of these roles. Moreover, by classifying the space of potential actions and characterizing these actions, a catalog of action patterns can provide a common language for referring to potential actions, and can provide opportunities for their systematic analysis and comparison. As we try to design CASTs that require attentive, mindful engagement with information, some researchers are highlighting the importance of careful study of the transactions that users make as they interact with these tools (e.g., Kim and Reeves, 2007; Brey, 2005; Dascal and Dror, 2005; Thomas and Cook, 2005). A framework such as EDIFICE-AP can provide investigators with a systematic support structure for thinking about these transactions. Without frameworks that organize and characterize fundamental aspects of the interaction design space, the approach to both research and practice must be largely ad-hoc and rely mostly on personal anecdotes and intuition.

Information and Information Space

Information can originate from many different sources (Bates, 2006). These sources can be concrete (e.g., a molecule), existing within a physical space, or abstract (e.g., financial markets), originating from a non-tangible, non-perceivable source. An information space is an environment, source, domain, place, or area of containment from which a body of information originates. The concept of information does not yet have a universally agreed-upon definition, and is defined in different ways depending on the context in which it is used (Marchionini, 2010). We adopt Bates’ (2005, 2006) definition of information—that information is the pattern of organization of matter and energy—e.g., physical objects, energy fields and forces, conceptual structures, and semantic relationships. This definition of information is broad and encompasses all visible, invisible, concrete, and abstract organizational patterns and sources—micro entities (e.g., DNA structure of a cell), hard-to-reach entities (e.g., rocks on distant planets), and non-physical entities (e.g., scientific concepts). Information by itself does not have inherent meaning. Meaning must be assigned to it (Stonier, 1990). For instance, electromagnetic waves travelling through space have no meaning until they are interpreted in a contextual setting. As such, giving meaning to information and integrating it into other pre-existing mental forms is an essential feature of any complex cognitive activity (Bates, 2005; Sternberg and Ben-Zeev, 2001).
When performing complex cognitive activities, users often need to access and combine information from different sources. For instance, an analyst reasoning about a financial event may need to access financial records, historical reports, legal information, and social or business networks. As CASTs can mediate access to any blending of different sources of information in a seamless manner, in this paper the term information space refers to any source of information, whether simple and from a single domain or complex and spanning multiple sources. It is important to note that in some research areas, the term information space refers to a dataset or the data records in a database. In this paper, however, we use the term in its broadest sense, encompassing datasets and data records—whether structured or unstructured, homogeneous or heterogeneous, dynamic or static—as well as web logs, images, videos, text documents, and any other item or collection of items that contains information. For instance, an analyst could be working with multiple datasets, streams of incoming data, unstructured text documents, audio and video recordings, and photographs, all of which are contained within the information space with which the analyst is concerned.

### Visual Representations

Because CASTs are computational environments, all information within them, whether originating from concrete or abstract spaces, needs to be visually encoded in a representational form at the interface of the tool to be accessible to users. Therefore, a representation acts as a perceptible form within which an information space’s items are encoded. Consequently, the representation, acting as a mental interface, can connect the human mind to the information space. In this paper, we refer to external representations displayed at the interface of CASTs as visual representations (VRs). Instances of VRs include diagrams, maps, photographs, glyphs, tables, scatter plots, node-link trees, text, and videos. Although VRs give information a tangible form, making it accessible at the interface, they seldom encode the totality of an information space. VRs usually encode only a subset of an information space. To provide an example, an information space may consist of climate data from a 100-year period. A given VR would be unlikely to encode the whole space (information regarding temperature, humidity, rainfall, atmospheric pressure, meteorological measurements, and their trends, outliers, cycles, and changes, for example), but would be likely to encode only some subset of the whole space, such as trends in global temperature change or the relationship between temperature change and CO2. When VRs are made interactive, however, users can act upon them to alter the manner in which information is encoded, such as by encoding hidden information, hiding encoded information, or interactively new information.

In order to design, analyze, and evaluate VRs systematically, a common conceptualization and vocabulary is required. Moreover, to operate at a foundational level, it must account for all kinds of different VRs, whether used for educational, financial, scientific, or other purposes, in a logical and consistent manner. Towards this end, a useful lens through which VRs can be viewed is that of general systems theory. This theory analyzes structures and properties of all systems at a general level, regardless of the particular form or domain of application (Skyttner, 2005). Generally speaking, a system is an organized whole composed of parts that generate emergent properties through their interrelationships. VRs are organized wholes (e.g., treemaps, radial diagrams), composed of parts (e.g., encodings and visual marks) that generate (i.e., communicate) emergent properties of an information space (e.g., patterns, correlations). As VRs can be considered as systems, general systems theory can therefore serve as a foundation upon which a science of VRs can be built. Essentially, all systems are hierarchical in nature, and are composed of layers of sub-systems (also referred to as entities, elements, objects, components, or parts, depending on the level of analysis) that have properties and relations with one another. It is the relations among entities at one hierarchical level that give rise to emergent properties at the level above. Accordingly, viewing VRs as systems allows for their discussion and analysis at different hierarchical levels in a systematic fashion. Any VR that is not simple and atomic can be decomposed into a set of sub-VRs (i.e., sub-systems), each of which can be further decomposed into other sub-VRs, all the way down to the atomic level of the VR in which information items are encoded as simple visual marks. Using general systems theory, researchers and designers can not only discuss the structure of VRs and how their sub-systems relate to communicate emergent features of an information space, but can also discuss interaction design in a precise manner. If a VR has a particular number of sub-VRs at different hierarchical levels, for instance, designers can think about sub-VRs with which a user should be able to interact, how such interaction should take place, how the overall state of the system will be affected in terms of its entities, properties, and relationships, and how emergent features of an information space are communicated.

### Complex Cognitive Activities, Tasks, Actions, and Events

Similar to VRs, complex cognitive activities can be regarded as hierarchical and emergent in nature (Funke, 2010). In the context of CASTs, complex cognitive activities emerge from lower-level tasks, which emerge from lower-level actions, which emerge from lower-level events. In addition, each level may be classified at finer levels of granularity: a complex cognitive activity may include sub-activities, a task may include sub-tasks, and so on. For instance, the activity of triaging a set of documents to find out whether they are semantically related may be comprised of lower-level tasks such as scanning the documents, extracting information, building associations among similar information items, and comparing these items. The task of extracting information may involve such actions as selecting a document, opening it, navigating it, selecting some items in it, and copying some items from it. Each of these actions in turn can be implemented in many different ways and using different input techniques, all the way down to low-level
events, such as mouse-clicks or gestures and touches at the physical level of the interface (see Sedig et al., 2013; Sedig et al., 2012a for a more detailed discussion of these different levels). In this paper, we are mainly concerned with actions and how different actions enable and facilitate higher-level tasks and activities.

To develop a more adequate understanding of how actions influence the performance of complex cognitive activities, particular activities must be identified and characterized. Researchers and practitioners require a sense of the characteristics of activities to determine how actions can and should support them. Some of the main complex cognitive activities are: analytical reasoning, problem solving, planning, sense making, forecasting, knowledge discovery, decision making, and learning (Bransford et al., 2000; Fludes et al., 2006; Funke, 2010; Hogarth and Makridakis, 1981; Klein et al., 2006; Knauff and Wolf, 2010; LeBoeuf and Shafir, 2005; Leighton and Sternberg, 2004; Mason, 2002; Morris and Ward, 2005; Sternberg and Ben-Zeev, 2001). Although future research is needed to explicate these activities in the context of interaction design, we briefly characterize three of them (analytical reasoning, problem solving, and sense making) to demonstrate their particular characteristics.

**Analytical Reasoning.** Analytical reasoning is a special type of reasoning. Reasoning itself refers to an activity in which information is used to draw inferences or conclusions (Leighton, 2004). In other words, reasoning can be seen as a transformative process in which new information is derived from the old, given information (Gilhooly, 2004). Analytical reasoning is based on rational, logical analysis and evaluation of information. It is an umbrella term covering many different kinds of reasoning: inductive, deductive, analogical, probabilistic, hypothetico-predictive, heuristic, syllogistic, categorical, and others (Halpern, 2003; Leighton and Sternberg, 2004). Analytical reasoning is a core concern of visual analytics (Thomas and Cook, 2005). As opposed to other complex cognitive activities, such as sense making and knowledge discovery, analytical reasoning is a structured, disciplined activity. Moreover, it is usually an iterative and non-linear process that involves tasks such as determining which resources to use, tracing and identifying cause-effect relationships, assessing the state of an information space, predicting future states of an information space, asserting and testing key assumptions, testing biases, and identifying and assessing alternatives (Heuer, 1999; Thomas and Cook, 2005).

**Problem solving.** This activity is concerned with searching through an information space to discover a path that connects a current state of information to some desired, goal state (Newell and Simon, 1972). A problem is a gap between two information states that should be bridged. Due to human cognitive limitations with regard to the amount of information that can be processed in working memory, problem solving is often a step-by-step process of connecting a current state to a sub-goal and eventually reaching the desired goal (Morris and Ward, 2005; Thagard, 2000). Problem solving typically begins by constructing a mental representation of the information space—i.e., a set of possible states of the problem, the current state, and possible goal states—as well as identifying the possible actions that can be performed to bridge the gap between information states (Fischer et al., 2012). Problem solvers then use strategies to reach desired goals or sub-goals, which involves changing their internal, mental representations and/or changing external representations—i.e., VRs (Fischer et al., 2012). A common heuristic strategy is means-end analysis, in which the goal or sub-goal is compared to the current state, the difference between them is assessed, and an action is chosen to reduce the difference and to gradually bridge the gap between information states (Sternberg and Ben-Zeev, 2001).

**Sense making.** This activity is concerned with developing a mental model of an information space about which one has insufficient knowledge (Dervin, 1992; Klein et al., 2006). Sense making often involves a sequential process of performing tasks such as scanning the information space, assessing the relevance of items within the space, selecting items for further attention, examining them in more detail, and integrating them into mental models (Pirolli and Card, 2005). Other interlocking tasks and sub-tasks include discovering the space’s structure and texture (e.g., vocabulary, resources, missing items), establishing questions to be asked, determining how to organize the answers, searching for pieces of information, encoding information to answer task-specific questions, reducing operational costs, filtering aspects of information, and categorizing items of information (Qu and Furnas, 2005; Pirolli and Russell, 2011; Russell et al., 1993).

**Pragmatic vs. Epistemic Actions**

Kirsh and Maglio (1992, 1994) have identified two types of actions: pragmatic and epistemic. Pragmatic actions are taken to transform the external world to achieve a physical goal (e.g., cooking a piece of meat before eating it). Epistemic actions are taken to transform the world to facilitate mental information-processing needs (e.g., rotating a jigsaw piece to explore potential fit while solving a jigsaw puzzle). Epistemic actions, then, can play an important role in complex cognitive activities (e.g., for usage in planning and sense making, see Clark, 1998a; Liang and Sedig, 2010). Performing external epistemic actions on visible VRs or latent parts of an information space (which is stored in computer memory) not only change and alter the VRs, but also affect and shape the information-processing functions of users of a CAST and help set and define new goals (Brey, 2005; Kirsh, 1997).
Epistemic Action Patterns

Although cognitive scientists have made a distinction between pragmatic and epistemic actions, they are not clear about the level at which such actions take place. In other words, it is not clear whether such phenomena occur at the level of tasks, actions, or events. While it may not be important to make such distinctions for cognitive science research, as discussed above, such distinctions are important to provide clarity and precision in the context of CASTs. In this paper, we are concerned with epistemic actions at the level of individual action-reaction pairs, rather than at the level of tasks—which typically involve many actions and reactions, or at the level of events—which involve physical occurrences at the interface. That is, for the purposes of this paper, epistemic actions occur between these two levels. In this context, epistemic actions are those actions that are performed with CASTs to facilitate mental information-processing needs. In this paper we are interested in epistemic actions at a level of abstraction that is independent of their physical performance, the manner in which they are implemented, the techniques that may be used to perform them, the users who perform them, and the technologies and tools that mediate their performance. A pattern can be defined as a regularity in some dimension (Salingaros, 1999). An epistemic action pattern, then, is a regularity in terms of an action-based characterization and its utility in the context of performing complex cognitive activities, and not necessarily in terms of other characteristics such as implementation and technological platform. In this sense, an epistemic action pattern is one that has a timeless, invariant quality in supporting human cognitive activities (see the framework section for more discussion of patterns and their utility).

Interactive Coupling and Complex Cognitive Activities

Over the past few decades, cognitive science research has increasingly emphasized the distributed nature of cognitive phenomena (e.g., Brey, 2005; Hollan et al., 2000; Salomon, 1993; Zhang and Norman, 1994). The theory of distributed cognition states that cognitive processes are not solely the product of the inner functioning of the brain. Rather, they result from relationships between internal mental representations and the external environment. These relationships with the external environment take place at several levels: relationships with culture, society, other individuals, computational artifacts, and external representations. Cognitive functions are, hence, emergent phenomena taking place across the brain, body, and these aforementioned levels. As such, the external environment aids the mind, becomes coupled with it, and can extend it (Clark, 1998a). When using CASTs, cognitive processes emerge from a coupling that is formed between the internal representations and processes of the user and external representations and processes at the interface (Kirsh, 2009). Numerous sources suggest that external representations and actions play an important role in facilitating the performance of all complex cognitive activities. For instance, VRs can facilitate learning (Greeno and Hall, 1997), and acting upon them is important in learning (Burdea and Coiffet, 2003; Cairncross and Mannion, 2001; Cobb and Fraser, 2005; Rogers and Scaife, 1997). The same is true of planning (Cox and Brna, 1995; Neuwirth and Kaufer, 1989; Morris and Ward, 2005); problem solving (Jonassen, 2003; Zhang, 2000); decision making (Beach and Connolly, 2005; Kleinmuntz and Schkade, 1993); sense making (Qu and Furnas, 2005; Sedig et al., 2005b); and knowledge discovery (Fayyad et al., 2002).

All the above activities involve processes through which VRs are decoded, linked, coordinated, and harmonized in the pursuit of reaching new goals and conclusions (Leighton, 2004). However, there are some factors that interfere with the proper execution of these activities, such as working with poorly-designed VRs, not having adequate mechanisms for manipulating and transforming the VRs, and not having appropriate ways for combining and integrating different VRs (Sloman, 2002). Another compounding factor is that people may see the same VR differently: some may see more detailed configurations of it, while others may see its more abstract structure (ibid.). At a basic level, complex cognitive activities involve the performance of simple visual sub-tasks, such as identifying an item or locating two close information items. However, even with the best-designed VRs, beyond the performance of simple tasks, the form, structure, amount, degree of abstraction, complexity, density, and other properties pertaining to how information is encoded affect the quality and process of complex cognitive activities (Anderson et al., 2002; Blackwell et al., 2004; Hegarty et al., 2002; Parsons and Sedig, in press; Peterson, 1996; Shah and Miyake, 2005; Zhang and Norman, 1994). The features of a VR can create a perceptual and cognitive distance between the VR and a user’s mental processes. This distance needs to be bridged for people to carry out mental activities using the VR. During these activities, it may be necessary to switch from one form of observation to another. Because processing VRs in the mind is not easy (Sloman, 2002), to support their mental activities, humans tend to externalize their mental processes by externally acting upon VRs (Clark, 2008; Kirsh, 2009; Sedig, 2009). As cognitive processes are intrinsically temporal and dynamic, interactive VRs potentially create a harmony and a tight temporal coupling with cognitive processes (Kirsh, 1997; 2005). As part of this dynamically coupled cognitive system, the user and the CAST each have a causal influence—in other words, the user and the CAST are continuously affecting and simultaneously being affected by each other (see Clark, 1998b).

Brey (2005) suggests that the distributed coupling between a user and a tool can be weak or strong. In the case of weak coupling, the external aids are usually tools (e.g., representational or physical) that do not actively participate in the information-processing functions of the mind. These tools do not necessarily need to be static. They can be dynamic, but not inviting of explicit action choices for human participation. In distributed cognitive phenomena,
interactive engagement with an external tool can strengthen the coupling and create a dialogical relationship with it. In other words, in distributed complex cognitive activities, interaction can make the coupling stronger. Unlike ordinary representational or physical tools, since CASTs are made of interactive visual representations, they can provide stronger coupling in that they serve the mind by being more than just externalizers of information spaces. In addition to their representational function, they can have built-in designed choices and conditional algorithmic behaviors that allow their users to engage in active, elaborative participation. These special external actions can be performed to support information-processing functions and complex cognitive activities—i.e., rather than performing “in-the-head operations,” external actions provide emergent “operational capabilities” (Clark, 1998a).

THE EDIFICE-AP FRAMEWORK

The presentation of EDIFICE-AP is divided into 5 sections: First, some of the characteristics of the framework will be examined. This includes a discussion of how such characteristics address many of the research needs previously identified. Second, the methodology for devising EDIFICE-AP will be examined. Third, EDIFICE-AP’s catalog of action patterns will be presented. Thirty-two epistemic action patterns are identified and characterized. For ease of reading, only four of the action patterns are elaborated upon in this section—that is, their utilities in supporting different complex cognitive activities are discussed, and examples of CASTs from different domains in which such actions have been and can be used are given. Readers are referred to Appendix A for detailed discussion of the additional 28 actions. Fourth, a scenario involving a sense making activity is used to demonstrate how EDIFICE-AP can help with the systematic analysis and design of epistemic actions in CASTs. Finally, in light of these four sections, some existing work is discussed to demonstrate how EDIFICE-AP is unique and novel.

Characteristics of EDIFICE-AP

As was mentioned in the introduction, EDIFICE-AP is intended to achieve a number of goals that address extant research needs: 1) to be syncretic, unifying a number of previously disconnected ideas into a coherent theoretical model; 2) to be general, operating at a level of abstraction that is applicable to all kinds of technologies, activities, users, and VRs; 3) to be comprehensive, identifying patterns that cover an extensive range of actions; and, 4) to be generative, possessing the ability to motivate design creativity as well as to stimulate further theoretical and empirical research. This section will elaborate upon each of these characteristics.

Syncretic

EDIFICE-AP is not simply a list of actions; rather, it unifies and integrates a number of ideas that are often discussed in isolation, in a logical and coherent manner, in order to provide a theoretical foundation that informs the conceptualization of human-information interaction in the context of complex cognitive activities.

As was discussed earlier, actions can be divided into two types: epistemic and pragmatic. Epistemic actions—those related to knowledge and knowing—are the concern of this paper. Just as different physical tools can extend one’s reach into a physical space, the action choices offered by a CAST can extend the human mind, like tentacles, to reach into an information space to perform operations upon it, such as by bringing into view portions of the information that have not been encoded and displayed by the VRs at the interface level, or by viewing information from different perspectives, or by reorganizing the information (see Figure 1).

Unlike simple structured tasks, complex cognitive activities do not usually follow a programmed, recipe-like model (Clark, 1998a; Thomas and Cook, 2005). Individuals deploy general, high-level strategies to operate upon an information space. In this process, they actively perform all kinds of epistemic actions upon the external environment to help them alter it, and as a result, transform and support their own cognitive functions to gradually achieve the overall goals of the activity. Therefore, complex cognitive activities emerge at a macro level while actions are occurring at a lower level. A sequence of epistemic actions creates a chain that represents the trajectory for the emergence of the macro-level activity (see Figure 2).

The sequence of epistemic actions that allows users to carry out an information-based complex cognitive activity can be conceptualized as an epistemic cycle. Figure 3 depicts this cycle and its categorization into five spaces: information, computing, representation, interaction, and mental space (see Sedig et al., 2012a for more elaboration on these spaces and their relationships). Information comes from some space or spaces and must be stored within a CAST. This aspect of a CAST can be conceptualized as computing space—the place where information is processed, stored, and prepared. This space may involve data cleaning, fusion, filtering and other pre-processing procedures, as well as data mining, transformation, and other mathematical procedures. Information must then be encoded in visual form to be made perceptually accessible to users—this is done in representation space. The information made available in this space through VRs is typically only a subset of the total information that is available. In addition, this space is often comprised of VRs of items from information space as well as VRs of aspects
of interaction space (e.g., action possibilities and tools that are available to the user). The user perceives VRs and performs mental operations within mental space—the place in which internal mental events and operations (e.g., memory encoding, storage, and retrieval; apprehension; judgment; classification) take place. The user then selects an epistemic action from a set of available choices based on some overall epistemic goals and strategies, and acts upon VRs within representation space to effect some reaction. This space encompassing action and reaction can be considered as interaction space. The user then perceives the reaction, and the cycle repeats until the user is satisfied that a task or an overall activity is accomplished. In the context of design, allowing users to choose from a set of these epistemic actions means that designers must first know what kinds of actions exist and then build them into their designed CASTs.

Note: Vector art adapted from www.vectoropenstock.com under the Attribution Creative Commons 3.0 license

**Figure 1:** Action Choices Operating as Mental Tentacles to Reach into an Information Space

**Figure 2:** The Hierarchical Structure of a Complex Cognitive Activity and its Emergence over Time
EDIFICE-AP abstracts beyond the details of techniques to identify action patterns that are applicable to diverse activities (e.g., sense making, problem solving), domains (e.g., science, education, business, gaming), and users (e.g., analysts, learners, researchers). A number of characteristics can be identified that contribute to the general nature of EDIFICE-AP:

Action-pattern-based rather than technique-based. EDIFICE-AP identifies and characterizes epistemic actions as general patterns rather than as technology- or implementation-dependent techniques. Even though interaction techniques can also be patterns, they are typically characterized at a lower level and are often technology-dependent. For example, one of the epistemic action patterns identified in EDIFICE-AP is drilling. Drilling is a general pattern that refers to all instances of acting upon VRs to drill into them and get more detail about latent, interior information that is not perceptually accessible—that is, it is latent in the information space and has not been encoded at the interface level. This is a conceptual, pattern-based characterization of the action and its utility that is not concerned with how the action is carried out. However, designers can develop many techniques to enable users to perform this action. Examples of these techniques include mouse-over, right-clicks, spatial proximity, semantic zooming, gestures, and digital probes. No matter what technique is used, by applying it to a VR, some of its hidden and latent information can be displayed.

Making the distinction between epistemic action patterns at this level and techniques at a lower level is crucial for two reasons. First, there are many existing techniques already, with all kinds of names and characterizations, and many more that can be developed in the future. Organizing many techniques under the umbrella of an action-pattern-based characterization makes it much easier to navigate the landscape of design possibilities by making the number of action possibilities manageable. Second, techniques vary in how they are characterized. By unifying many of them under one pattern, designers and researchers can focus on the conceptual utility of the action and worry about...
techniques and implementations later. Therefore, given a deep information space, a subset of whose items has been encoded by a VR, it can be easily predicted that at some stage of interaction with the tool users may need to drill into the VR to access latent information. This knowledge makes it easier for the designer to provide users with such an action choice. Indeed, the designer may choose to provide users with different implementations of the same action pattern. The pattern names we have selected are very close to the dictionary definition of the actions in order to be suggestive of what the actions do. For instance, the action pattern ‘scoping’ suggests that the action deals with the range, extent, breadth, and/or scope of perception of an information space. Thus, pattern names suggest both the actions that may be needed as well as their epistemic utility.

Technology independent. The epistemic actions in EDIFICE-AP are independent of the technology through which complex cognitive activities are carried out. This is important—as technological platforms on which complex cognitive activities are performed change, EDIFICE-AP can remain resilient to these changes and still be applicable. Continuing with the drilling example, as new technological innovations (such as interactive tabletops, motion sensing input devices, virtual reality, augmented reality, and interactive surface projection environments) come into existence, these may result in the development of new techniques and methods for drilling into VRs. However, at a conceptual, general level, drilling will always exist as a distinct pattern of action with utility for performing complex cognitive activities, and designers will decide how to implement it using new technologies.

Activity independent. The epistemic actions in EDIFICE-AP are not directly linked to and dependent on the complex cognitive activities that they support. Since complex cognitive activities are emergent phenomena, combining different epistemic actions can result in countless trajectories of macro tasks and activities. Hence, whether the activity be sense making, decision making, planning, learning, knowledge discovery, or problem solving, a subset of the epistemic actions in EDIFICE-AP can be used to support it, depending on the contextual and situational needs of the activity and its users.

User independent. The epistemic actions in EDIFICE-AP can be performed by users of different ages and backgrounds. This is in contrast to techniques and implementations that can hinder some users from understanding how certain actions are carried out. For instance, while drilling, a young child may have difficulty doing a right-click and going down a menu to select an option. But the very same child may be able to use drilling techniques such as a moving a mouse cursor over an object or pressing an object on a touch-screen surface. Additionally, the actions in EDIFICE-AP can be used in different situations whether they be single user, collaborative, or multi-user settings. Since the actions are geared towards achieving activities, the activities can be carried out individually or collectively.

Processing-load independent. Since there is a joint epistemic partnership between the user and the CAST, when an epistemic action is performed its processing load is distributed across the mental, representation, interaction, and computing spaces—that is, between the CAST and the user (see Figure 3). This means that in some instances, the user may initiate an action, but most of the information processing load is carried out by the tool. It is up to researchers and the designers of the tool to determine how to distribute this load. This decision is dependent on several factors, such as the VRs that are used, the type of activity, and the users of the tool. For instance, in a CAST that is to be conducive to mindful reflection on the underlying relationships among the items in an information space when carrying out a learning activity, designers may need to let the users do most of the processing when performing an action (e.g., see Sedig et al., 2001; Liang et al., 2010). On the other hand, in a CAST that is designed to help users make time-critical decisions, for any given action there may be powerful algorithms and data mining features that shoulder most of the information processing load in the computing space.

Comprehensive

EDIFICE-AP is comprehensive in its scope. The 32 patterns that are identified are intended to cover the broad range of epistemic actions that are typically performed during sense making, problem solving, decision making, and other complex cognitive activities. The majority of interaction techniques, whether from information visualization, human-computer interaction, visual analytics, learning and knowledge technologies, digital libraries, or otherwise, are covered by EDIFICE-AP. However, although EDIFICE-AP is comprehensive, it is not necessarily exhaustive and may be expanded in the future.

Generative

EDIFICE-AP is generative in its nature. By providing a coherent conceptualization of the human-information interaction epistemic cycle and the emergent nature of complex cognitive activities, presenting a catalog of action patterns, discussing their utility for performing complex cognitive activities, examining how they have been used in some existing CASTs, and identifying potential usage scenarios, EDIFICE-AP can facilitate systematic design of CASTs and can be used as a reference to help with design decisions. In his paper discussing models for interaction design, Beaudouin-Lafon (2004) noted that interaction models that are generative help designers “create richer and more varied design spaces from which to develop innovative solutions” (p. 17). By identifying patterns of action that are not dependent on particular implementation details, designers gain a support structure that allows them to think
about the utility of an action, and then devise numerous innovative techniques and implementations that fit the particular context of use. In addition, because EDIFICE-AP is flexible and does not dictate any particular sequence of actions, designers can come up with different sequences of actions to be built into CASTs so as to most effectively support different tasks and activities. Furthermore, because the actions are identified in a conceptual, pattern-based fashion, designers can think about blending action patterns at a conceptual level to create new techniques. For instance, in many activities it may be beneficial to blend together the drilling and comparing action patterns. As a user acts upon two VRs, latent information is encoded and made visible while simultaneously identifying the degree of similarity between the two VRs. Another useful blending of patterns is that of sharing and cloning. A user may wish to share a VR to be used by a research team, for instance, but still keep a copy of the original. Indeed, action patterns can be blended in innumerable ways, each of which has distinct utility depending on the context of use. EDIFICE-AP is generative not only in the context of design; rather, its novel characteristics can also stimulate further theoretical research, and can motivate empirical studies that further examine the cognitive utility of the identified action patterns.

Methodology

This section describes the methodology for the construction of EDIFICE-AP, particularly in the context of achieving the desired characteristics discussed in the previous section. We describe the methods of achieving the following characteristics: 1) syncretic, 2) general, 3) comprehensive, and 4) generative, as well as the rationale and approach to the 5) characterization of interaction, action patterns, and VRs, 6) classification of action patterns, and 7) validity of action patterns. These do not signify a series of sequential steps; rather, they are interwoven and complementary aspects of EDIFICE-AP's development.

**Syncretic.** To develop a coherent theoretical foundation for EDIFICE-AP, we have reviewed literature from numerous disciplines, including information science, cognitive and learning sciences, information systems, cognitive technologies, information behavior, information visualization and visual analytics, library science, human-computer interaction, computer science, psychology, and philosophy of mind. Observations made during this review suggested that there are deep connections among these different disciplines. Thus, we have identified relevant and related ideas from these different areas—e.g., epistemic vs. pragmatic actions from cognitive science; distributed cognition and extended mind theory from cognitive science and philosophy of mind; events, actions, tasks, and activities from information science, information behavior, and psychology; complex cognition and cognitive activities from learning sciences, information science, and cognitive science; interaction from cognitive and knowledge technologies and human-computer interaction; and, encoding and representation of information from psychology, information visualization, and visual analytics. By seeking out commonalities in these different research areas, important insights into how humans interact with information to perform complex cognitive activities can be gained. Thus, this aspect of the development of EDIFICE-AP represents a conscious attempt to syncretize related but underdeveloped and fragmented ideas into a coherent theoretical whole to inform the conceptualization, design, and evaluation of CASTs.

**General.** In their comprehensive and critical review of human-computer interaction pattern languages, Dearden and Finlay (2006) suggested that an arguable weakness in many interaction design patterns is that they are strongly based on particular and current user interface paradigms, platforms, and/or technologies, and therefore do not embody a ‘timeless quality’ that is a necessary characteristic of good design patterns. Furthermore, they suggest that it is relatively easy to observe phenomena which could be put into a pattern-like form, but much more difficult to use these observations to develop and explicate good patterns. As Fincher (1999) noted, “practice can be captured at any scale, but it is the combination of capture and abstraction that makes the presentation of the ideas coherent” (p. 339). During the development of the action-patterns portion of EDIFICE-AP, we have attempted to avoid such pitfalls and to identify patterns at a consistent level of abstraction that is useful for both research and practice. Furthermore, as discussed elsewhere, we have made a conscious effort to generalize beyond particular tools, tasks, domains, users, and technologies to contribute to a general theoretical framework for human-information interaction in complex cognitive activities.

**Comprehensive.** The intention to generalize by seeking similarities through abstraction had an influence on the development of EDIFICE-AP in general as well as on the process of pattern identification in particular. Generally speaking, patterns for design are derived empirically from observations, rather than from first principles (Salingaros, 2000). One commonly used method of identifying patterns is the process of ‘pattern mining’—extracting patterns by observing previous designs. This process is used in many fields, including software design, architecture, and interaction design (e.g., Gabriel, 1996; Meszaros, 1996; Iacob, 2011). To identify the 32 action patterns that are part of EDIFICE-AP, we took a twofold approach. First, we analyzed 130 existing tools that are used to support complex cognitive activities in different domains. These included: 20 educational tools, 40 data and information visualization tools, 20 visual analytics tools, 10 productivity tools, 20 digital games, 10 decision support tools, 5 digital library tools, and 5 personal information management tools. The second approach was to conduct an extensive analysis of literature that presented new interaction techniques, surveyed existing techniques, or provided interaction taxonomies and catalogs. During this analysis, we identified and recorded common characteristics and utilities of techniques. In order to identify fundamental action patterns, we abstracted beyond the details of each technique and implementation to categorize them according to fundamental features that were not dependent on particular tools, platforms,
domains, or users. Although we do not claim that EDIFICE-AP’s action catalog is absolutely exhaustive, we do believe that it is comprehensive in that it accounts for the majority of interaction techniques that users perform in all of the aforementioned activities and domains.

**Generative.** As mentioned previously, by identifying patterns of action that are not dependent on particular implementation details, designers gain a support structure that allows them to think about the utility of an action, and then devise numerous innovative techniques and implementations that fit the particular context of use. One method we used for accomplishing this desired characteristic was to develop usage scenarios for each action pattern (see Appendix A). It is well known from research on human creativity that new ideas are often generated through novel combinations of old ideas and access to new information (Lau, 2011). By devising usage scenarios, EDIFICE-AP provides designers with different contexts in which action patterns can be implemented, each of which may trigger mental associations and generate new design ideas. To position EDIFICE-AP to stimulate further research, we have proposed high-level models and ideas (e.g., emergence of complex cognitive activities using CASTs, distribution of information processing across different spaces, effects of action patterns) that require further theoretical and empirical research to more fully explain and describe their features in different contexts. In addition, we have explicitly suggested a number of future lines of research that may be undertaken to develop a better understanding of how to design and evaluate CASTs (see summary and future work section).

**Characterization of interaction, action patterns, and VRs.** To address the issue of ambiguity in existing literature, interaction in EDIFICE-AP is characterized as a complex phenomenon that must be categorized into multiple levels to discuss it in a coherent and meaningful fashion. In this paper, we are chiefly concerned with interaction at only one such level: that of individual actions performed by a user and the subsequent reactions from a CAST. Each action pattern is characterized in light of this categorization. In addition, much effort has been made to remain consistent in characterizing each action pattern at the same level. Furthermore, the names of all action patterns contain the suffix ‘ing’, suggesting that users are the ones initiating the action. On a separate but related note, in an attempt to bring more accuracy and precision to interaction design, we have used general systems theory to characterize VRs. This allows for thinking about and discussing interaction design in a precise manner. Existing research is often not clear about what the object of an action is—it is often suggested that users act upon VRs, for example, but there is no specificity with regard to what portion of a VR is receiving the action. As VRs can be quite large and complex, this lack of precision can be problematic for research, design, and evaluation. Through the lens of general systems theory, any interface can be analyzed into its constituent components in a consistent manner. If VRs are conceptualized as systems comprised of sub-systems at multiple hierarchical layers, designers and researchers can think about sub-VRs with which a user should interact, how such interaction should take place, and how the overall state of the system will be affected. Although this portion of EDIFICE-AP’s characterization is important, it is not fully developed in this paper and requires explication elsewhere. Keeping it in mind, however, can help bring more accuracy and exactness to interaction design.

**Classification of action patterns.** During examination of the literature and existing CASTs, we observed a pattern in the way that actions were and could be implemented in relation to one another: there are some action patterns in which an action is performed in one direction and there is no natural opposite action. After committing such an action, users can only typically reverse it by performing an ‘undo’ action. On the other hand, there are some patterns that are natural opposites of one another—when an action is performed, there is another natural opposite action. Therefore, to bring more order and clarity to the interaction design space, the action patterns in EDIFICE-AP have been classified into unipolar and bipolar action patterns. It was further observed that in many existing CASTs the bipolar patterns appear together. Thus, such a classification is useful not only for research and design, but also for evaluation. For instance, an evaluator can use this classification to determine whether two bipolar actions do or should appear together. Such a classification also helps us to see that some actions do not have a natural opposite. This classification is not the only valid one; it is possible that other classifications may be useful for different users, tasks, and contexts.

**Validity of action patterns.** The validity of the action patterns can be assessed from two angles: an ontological one and an empirical one. First, each action pattern has ontological validity. After characterizing each pattern, we give examples of several complex cognitive activities that are supported. Furthermore, in Table 1 as well as in Appendix A, we provide examples of CASTs in which each action pattern is implemented. This is intended to demonstrate and validate the existence and necessity of the pattern. Therefore, there is no need to perform experiments to find out whether the action pattern actually exists. What is in need of experimentation, however, is the role that the action pattern has in supporting complex cognitive activities. We have provided evidence supporting the empirical validity of the effects and utility of the patterns by noting relevant studies that have been previously conducted. These research studies, dealing with the cognitive and epistemic roles of actions in the performance of complex cognitive activities, are from diverse bodies of literature concerned with psychology, learning sciences, human-computer interaction, information science, computer science, and cognitive and knowledge technologies. The fact that we have gathered empirical validation, however, does not obviate the need for further precise studies that more fully explicate the effects and utility of the action patterns.
Catalog of Action Patterns

In Thomas and Cook's (2005) research agenda for visual analytics, they called for the development of "a science of interaction" and the need for "a deep understanding of the different forms of interaction and their respective benefits" (73, italics added). They further stated that the "grand challenge of interaction is to develop a taxonomy to describe the design space of interaction techniques that supports the science of analytical reasoning" (76, italics added). EDIFICE-AP presents a catalog of 32 epistemic action patterns that describe the interaction design space at the action-reaction level of human-information discourse. Table 1 lists all the epistemic action patterns in EDIFICE-AP, briefly characterizes each, and identifies some CASTs in which each action pattern is implemented. Following this, four patterns are characterized and the utility of each for performing complex cognitive activities is discussed in detail. These four are scoping, translating, collapsing, and expanding. For ease of reading, characterizations of and discussions about the rest of the patterns are appended (see Appendix A).

Scoping

Characterization: Acting upon VRs to dynamically work forwards and backwards to view their compositional development and growth, either temporally or spatially. The term scope here is meant to signify the range, breadth, field, or amount of compositional information in view.

Utility: Among others, this action facilitates sense making, reasoning, investigating, and understanding (Chen and Morris, 2003; Chen, 2004; Card et al., 2006; Morey and Sedig, 2004; An et al., 2001). There are many situations in which users may want to discover the process or sequence of growth, development, or construction of an information space. Scoping can be useful for reasoning about the growth process of most complex phenomena, such as fractals, 3D structures, proteins, economic trends, and galaxies. Scoping the growth of such information spaces from elemental parts to more aggregate wholes can facilitate deeper understanding of the emergence of complex phenomena (Kaandorp, 1994). In general, the ability to analyze ideas by reasoning forward and backward and observing how information items are chained together is important in analytical thinking (Shrinivasan and Wijk, 2008). Also, in many circumstances, being able to explain the prevalence of given structures within an information space depends upon identifying the temporal order in which relations occur (Moody et al., 2005). Examples of such information spaces are co-citation networks (see Chen and Morris, 2003; Chen, 2004) and social networks (see Moody et al., 2005). Working with static VRs of such information spaces can lead to false interpretations compared to when a user is able to act upon VRs to see the temporal growth (Moody et al., 2005). Therefore, scoping VRs of network-like information spaces can aid in investigating relationships, observing trends, and understanding how clusters are merged and split over time (Toyoda and Kitasugawag, 2005). Moreover, in the context of research networks, understanding the evolution of a network can support the activity of forecasting research trends and studying a scientific community's life span (An et al., 2001).

An example of a CAST that implements the scoping pattern is Polyvise (Morey and Sedig, 2004), a tool for exploring the compositional structure and formation of complex 4D mathematical shapes. Figure 4 shows four successive stages of a scoping action being performed. By providing this action possibility, Polyvise allows users to gradually construct or deconstruct the VR to make reasoning about its composition more tractable. The temporal coupling that is formed between the user’s mental space and the VR, through such interaction, can facilitate the development of an accurate mental model of the information space.

Translating

Characterization: Acting upon VRs to convert them into alternative informationally- or conceptually-equivalent representations, each requiring different degrees and kinds of cognitive and perceptual processing.

Utility: Among others, this action facilitates problem solving, decision making, learning, reasoning, sense making, and understanding (Yi et al., 2007; de Jong et al., 1998; Spence, 2007; Peterson, 1996; Gotz and Zhou, 2008). In general, the effective acquisition and growth of knowledge depends on the use of appropriate representational forms (Peterson, 1996). However, the appropriateness of VRs depends on many factors related to the characteristics of the users, their tasks and activities, and the properties of VRs themselves (see Larkin and Simon, 1987; Peterson, 1996; Sedig and Liang, 2007). In the context of problem solving, sometimes one’s understanding of a problem while working with a certain VR is poor, and translating to another VR of the same problem leads to insight (Robertson, 2001; Anderson, 2000). This is because inferential abilities are fundamentally affected by external representations (Cox and Brna, 1995; Larkin and Simon, 1987; Kaput, 1989). Translating can allow learners, for instance, to see relationships between the parts of the problem and understand its underlying structure, and can open up new paths through a problem space (Robertson, 2001). In one study, Bodner and Domin (2000) noticed that problem solvers who translated information into alternative representations were the most successful in making sense of concepts in organic chemistry. One benefit of a translating action, in the context of interactive VRs, is that users can go back-and-forth to compare and contrast alternative VRs to assimilate different aspects of an information space into their mental
structures and increase understanding (see Spiro and Jehng, 1990; Godshalk et al., 2004). Overall, the translating action pattern has a high degree of utility for all activities, tasks, and users, since each informationally- or conceptually-equivalent VR enables different inferential abilities and reveals and emphasizes different aspects of an information space.

Table 1: Catalog of Epistemic Action Patterns

<table>
<thead>
<tr>
<th>Action</th>
<th>Characterization acting upon VRs to …</th>
<th>Example CASTs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annotating</td>
<td>augment them with additional visual marks and coding schemes, as personal meta-information</td>
<td>GeoTime (Eccles et al., 2008); Mendeley; Tableau</td>
</tr>
<tr>
<td>Arranging</td>
<td>change their ordering, either spatially or temporally</td>
<td>Table Lens (Rao and Card, 1994); InfoZoom</td>
</tr>
<tr>
<td>Assigning</td>
<td>bind a feature or value to them (e.g., meaning, function, or behavior)</td>
<td>NetLogo (Wilensky, 1999)</td>
</tr>
<tr>
<td>Blending</td>
<td>fuse them together such that they become one indivisible, single, new VR</td>
<td>Microsoft Word</td>
</tr>
<tr>
<td>Cloning</td>
<td>create multiple identical copies</td>
<td>Cytoscape (Shannon et al., 2003)</td>
</tr>
<tr>
<td>Comparing</td>
<td>determine degree of similarity or difference between them</td>
<td>Multidatex (Wu et al., 2006); Panopticon</td>
</tr>
<tr>
<td>Drilling</td>
<td>bring out, make available, and display interior, deep information</td>
<td>OECD eXplorer, Sunaeneon</td>
</tr>
<tr>
<td>Filtering</td>
<td>display a subset of their elements according to certain criteria</td>
<td>Film Finder (Ahlberg and Shneiderman, 1994), Tableau</td>
</tr>
<tr>
<td>Measuring</td>
<td>quantify some items (e.g., area, length, mass, temperature, and speed)</td>
<td>GeoGebra</td>
</tr>
<tr>
<td>Navigating</td>
<td>move on, through, and/or around them</td>
<td>K-Lattice Machine (Sedig et al., 2005a); CGV (Tominski et al., 2009)</td>
</tr>
<tr>
<td>Scoping</td>
<td>dynamically work forwards and backwards to view compositional development and growth</td>
<td>Polyvise (Morey and Sedig, 2004); Gephi (Bastian et al., 2009)</td>
</tr>
<tr>
<td>Searching</td>
<td>seek out the existence of or locate position of specific items, relationships, or structures</td>
<td>Health Infoscape; Vizster (Heer and Boyd, 2005)</td>
</tr>
<tr>
<td>Selecting</td>
<td>focus on or choose them, either as an individual or as a group</td>
<td>EdgeMaps (Dörk et al., 2011); Dust &amp; Magnet (Yi et al., 2005)</td>
</tr>
<tr>
<td>Sharing</td>
<td>make them accessible to other people</td>
<td>Tableau, Mendeley</td>
</tr>
<tr>
<td>Transforming</td>
<td>change their geometric form</td>
<td>Vizster (Heer and Boyd, 2005)</td>
</tr>
<tr>
<td>Translating</td>
<td>convert them into alternative informationally- or conceptually-equivalent forms</td>
<td>Archim, Panopticon</td>
</tr>
<tr>
<td>Accelerating/Decelerating</td>
<td>increase or decrease speed of movement of their constituent components</td>
<td>OECD eXplorer; NetLogo (Wilensky, 1999)</td>
</tr>
<tr>
<td>Animating/Freezing</td>
<td>generate or stop motion in their constituent components</td>
<td>Gapminder, Step</td>
</tr>
<tr>
<td>Collapsing/Expanding</td>
<td>fold in or compact them, or oppositely, fold them out or make them diffuse</td>
<td>VisANT (Hu et al., 2009); Gephi (Bastian et al., 2009)</td>
</tr>
<tr>
<td>Composing/Decomposing</td>
<td>assemble them and join them together to create a new, whole VR, or oppositely, break whole entities up into separate, constituent components</td>
<td>Hyperchem, SmartJigsaw3D (Ritter et al., 2000)</td>
</tr>
<tr>
<td>Gathering/Discarding</td>
<td>gather them into a collection, or oppositely, throw them away completely</td>
<td>Microsoft Onenote, Gephi (Bastian et al., 2009)</td>
</tr>
<tr>
<td>Inserting/Removing</td>
<td>interject new VRs into them, or oppositely, get rid of their unwanted or unnecessary portions</td>
<td>Visible Body, ModellingSpace (Avouris et al., 2003)</td>
</tr>
<tr>
<td>Linking/Unlinking</td>
<td>establish a relationship or association between them, or oppositely, dissociate them and disconnect their relationships</td>
<td>MindJet; DEMIST (Ainsworth and van Labeke, 2001)</td>
</tr>
<tr>
<td>Storing/Retrieving</td>
<td>put them aside for later use, or oppositely, bring stored VRs back into usage</td>
<td>HARVEST (Gotz et al., 2010); Cytoscape (Shannon et al., 2003)</td>
</tr>
</tbody>
</table>
Figure 4: An implementation of the Scoping Pattern: Acting upon a VR of a 4D Shape to Explore its Compositional Structure

An example of a CAST that implements the translating pattern is Gapminder. Figure 5 shows two different states of the interface during the performance of an activity. In Figure 5 (L), the scatterplot VR depicts the relationship between life expectancy and GDP per capita for a number of countries. Such a VR has certain benefits, such as facilitating the perception of anomalies, deviations, and outliers, and supporting the performance of complex cognitive activities that involve reasoning about trends and patterns within an information space (Parsons and Sedig, 2013a). Although each circle in the scatterplot represents a country, the location of which is encoded by color and corresponds to the map in the upper-right portion of the interface, the user may find it useful to translate the VR to the map-based VR shown in Figure 5 (R). In other words, although both VRs have very similar information content and are conceptually equivalent, certain tasks and inferences may be much more tractable using one VR over the other. Therefore, by providing the ability to translate the VR, the CAST can more effectively support the cognitive and contextual needs of its users.

Note: Free material from www.gapminder.org.

Figure 5: An Implementation of the Translating Pattern: Acting upon a VR to Convert it From One Form to Another

Collapsing/Expanding

Characterization: Acting upon VRs to fold them in and/or make them compact, or oppositely to fold them out and/or make them diffuse.

Utility: Among others, these actions facilitate sense making, reasoning, exploring, and investigating (Abello et al., 2006; Noel and Jajodia, 2004; Pinzger et al., 2008). Both folding and expanding representations are important in complex cognitive activities. Such actions can facilitate making sense of relationships among information items in complex information spaces by reducing and increasing detail when performing tasks (Pinzger et al., 2008). VRs with high degrees of density and/or complexity can place a large burden on users' perceptual and cognitive faculties and thus negatively influence the performance of tasks and activities (Demetriadis and Cadoz, 2005; Pirolli et al., 2001). VRs that are too dense, for instance, can place an unmanageable amount of the information-processing load on a
user’s working memory (Green and Petre, 1996). VRs with a lower degree of complexity have been empirically shown to have more correct responses to tasks as well as better reaction times while identifying trends in information spaces (see Cruz-Lemus et al., 2010; Huang et al., 2009; Meyer et al. 1997). Collapsing can allow users to condense a set of items into one, thereby reducing complexity and/or density and facilitating the comprehension of overall relationships and trends. Expanding, on the other hand, allows users to explore information spaces in a more diffused, opened-up fashion with more detail (Noel and Jajodia, 2004; Abello et al., 2006). When encoding a complex information space, there is always a trade-off between displaying low-level detail and high-level structure, and it is generally useful to provide users with access to both. Accordingly, expanding can be used along with collapsing to facilitate quick movement through spaces (Dachselt and Ebert, 2001). When dealing with complex VRs, expanding areas of interest while keeping other areas folded in helps manage information overload (Samp et al., 2008).

An example of a CAST that implements the collapsing and expanding patterns is VisANT (Hu et al., 2009), a tool that supports the exploration of protein complexes. In Figure 6, the representation space of VisANT is quite complex, as there are many items and relationships between them. Depending on the context, as discussed above, this may hinder the performance of perceptual and cognitive tasks. By implementing the collapsing pattern, VisANT provides users with the ability to act upon VRs to collapse them in order to reduce complexity and to make sense of higher-level relationships in the information space. Doing so may allow users to make sense of clusters within the information space and identify major pathways between them. Users may need to repeatedly collapse (e.g., Figure 6 L to R) and expand (e.g., Figure 6 R to L) different portions of the VR to accomplish various tasks while performing a complex cognitive activity.

![Figure 6: An Implementation of the Collapsing and Expanding Patterns: Acting upon a VR to Fold In and Fold Out some of its Constituent Components](image)

**Integrated Scenario: A Sense Making Activity**

This section is intended to demonstrate how EDIFICE-AP can help with the systematic design and evaluation of epistemic actions in CASTs. A designer may identify characteristics of an information space and then consult EDIFICE-AP to become aware of relevant action patterns, which in turn can stimulate creativity in the design process. The designer can then implement desired actions in a systematic manner with their epistemic utility at the forefront of consideration. In a similar fashion, an evaluator may consult EDIFICE-AP to facilitate thinking about potential action patterns and subsequently assess how well a particular CAST is designed. The evaluator may also use EDIFICE-AP’s catalog as a support structure for comparing CASTs based on their provision of action possibilities to evaluate how well they support given complex cognitive activities. Additionally, designers and evaluators can ask questions based on the actions identified in EDIFICE-AP to determine which action(s) should be included in a given CAST. For example, “Do users need to be able to see the compositional development and growth of the information space, either spatially or temporally, to be able to perform the complex cognitive activity more effectively?” If the answer is ‘yes’ then the scoping pattern should be implemented in the CAST. A list of similar questions can be asked systematically, leading to decisions regarding the inclusion or non-inclusion of the other action patterns and their blending.

The scenario presented here involves a sense making activity, in which a financial analyst needs to make sense of stock market activity in the US. In such an activity, the user (she) has insufficient knowledge of the information space and needs to develop a clearer mental model of it. Through a cycle of actions, her conceptualization of the space...
gradually evolves such that she can eventually develop an adequate mental model of the space. The rest of this section demonstrates how using EDIFICE-AP and thinking about the combination and integration of a number of different action patterns can facilitate design and evaluation of a CAST that supports the sense making activity.

As discussed previously, complex cognitive activities are hierarchical and emergent, resulting from the combination and interaction of a number of sub-activities, tasks, sub-tasks, actions, and events. While making sense of large and complex information spaces, users perform different sub-activities and gradually synthesize them once adequate connections between pieces of information can be made. One sub-activity that the user would likely perform in this scenario is knowledge discovery—exploring the information space to discover useful patterns within it. This sub-activity may involve the user browsing the information space to identify the distribution and dispersion of stocks and to distinguish between different categories of stocks. The user would also likely need to perform the task of organizing some of this information not only by identifying stocks and distinguishing between them, but also by ranking them according to different criteria. Consider the actions discussed below and how they facilitate the performance of such tasks and gradually lead to the emergence of activities.

While the user is browsing and trying to identify prominent features of the information space, one action that can facilitate such a task is drilling. The user can repeatedly drill into different stocks or industries to identify properties such as open and close values, turnover, and market capitalization. As the user begins to get a general sense of some of the main items within the information space, she will likely wish to identify and distinguish between items according to some particular criteria. One action pattern that facilitates such a task is filtering. Figure 7 shows a treemap VR of the stock market information space. In Figure 7 (L), an overview of the information space is provided, with stocks categorized according to industry. Figure 7 (R), however, shows the result of the user filtering the VR to display only stocks with relative activity above 600%. Only a handful of stocks are now shown, and the user can easily identify stocks that have seen a very high amount of recent activity, which may stimulate hypothesis formulation, information searches, and outlier detection. The user may perform similar filtering actions according to other criteria such as market capitalization, turnover, and degree of change.

As the user progresses in the activity, she will need to organize all of the identified items to develop a richer mental model of the information space. One action pattern that can help in this regard is comparing. Figure 8 shows the user acting upon the VR to compare two stocks within the technology industry. Performance of this action allows the user to further distinguish between different stocks based on their properties, and to begin to develop an understanding of the rank and ordering of stocks based on their different properties. In addition, an action pattern that can further facilitate this task is that of arranging. By acting upon the VR to adjust its spatial arrangement, the user can easily perceive the ranking of all stocks based on their properties such as market capitalization and trading activity. Figure 9 depicts the user arranging the VR to reorder it according to market capitalization (L) and trading activity (R).

Although the treemap VR exploits certain perceptual features to facilitate tasks and activities, no one VR can sufficiently support all tasks and activities. Thus, a designer could predict that with such a complex information space the user would benefit from having access to different VRs of the same underlying information. A previously discussed action that has utility in most activities is translating. Figure 10 shows the user translating the treemap VR to an alternative form—a scatterplot VR. In this case, the information content of the scatterplot is very similar to that of the treemap. The form of the VR, however, exploits different perceptual features and facilitates tasks and activities in different ways than the treemap VR (see Parsons and Sedig, 2013a, for more discussion of the perceptual and cognitive utility of different VRs).
Figure 8: Comparing the Microsoft and IBM Stocks

Figure 9: Arranging the VR to Reorder it According to Market Capitalization (L) and Trading Activity (R)

Figure 10: Translating the Treemap VR (L) to an Alternative Form (R)
Consulting the action catalog provided by EDIFICE-AP can help designers and evaluators to identify other action patterns that can further support the user in the performance of her sense making activity. For instance, consider the following action patterns from EDIFICE-AP in the context of the current scenario:

**Gathering/Discarding.** While interacting with the treemap VR, the user may become interested in a few particular stocks. For example, she may be surprised at the large gap in market value between certain stocks that she thought would have a similar market value. As a result, she may gather them into a temporary collection for subsequent analysis. She may then discard some that are not pertinent to a particular task.

**Scoping.** In order to identify relationships and temporal trends among stocks, industries, or sectors within the information space, the user can act upon the VR by scoping it. For instance, she may wish to see the growth of the oil and gas sector over the past few decades, particularly around specific events such as the 1973 oil crisis. Providing mechanisms for dynamically moving forwards and backwards to see the temporal growth and development of the VR may facilitate such tasks.

**Navigating.** The user may wish to identify connections within the information space that are not visible in the VR. For instance, given the VR in Figure 8, she could navigate it by traversing its stocks according to market value. This could allow the user to identify the ranking of stocks while keeping the spatial layout of the VR consistent, so that she can identify their positions within the different sectors and industries.

**Cloning.** The user may wish to perform certain actions upon a VR, but may still wish to keep the original state of the VR. Additionally, she may wish to have both the new state and the original state simultaneously to compare or perform other tasks with them. As such, she can act upon one of the VRs in the plot that represents an individual stock to clone it and make a copy. She may then perform numerous actions on the cloned VR, such as assigning, annotating, or drilling.

**Assigning.** After cloning a VR of a stock, the user can assign a certain value or feature to the VR and perceive its effect. For example, she could assign a particular value to the stock to forecast how it may affect other stocks within the sector or to project its growth over a period of time.

**Annotating.** After cloning a VR and assigning certain properties to it, the user may want to make a record of what she has done, why she has done it, and any expectations, outcomes, or other observations that she has. If the CAST provides the ability to annotate VRs, she can act upon a VR to add such meta-information to it. Visiting the annotation at a later point in time may facilitate sense making and/or may provide insight into her thought processes.

**Storing/Retrieving.** At any point, especially after altering VRs by annotating, assigning, or cloning, the user may wish to store them. At some later point in time she can then retrieve them to continue with other activities or tasks.

Not only does EDIFICE-AP provide an action catalog that allows designers and evaluators to think about action possibilities in a systematic fashion, but it also provides a framework for thinking about the overall human-information discourse and the emergent nature of complex cognitive activities. More specifically, EDIFICE-AP helps in thinking about how actions allow users to mentally `reach into' and perform operations on an information space, how the continual occurrence of actions, reactions, and perceptions forms an epistemic cycle, and how complex cognitive activities emerge over time from the combination and interaction of actions and tasks (see Figures 1-3). Figure 11 demonstrates how the provision of different action patterns allows users to operate on represented information in different ways, and to `reach into' an information space to access new information or modify or remove existing information. Figure 12 depicts how the sense making activity discussed in this section emerges at different levels over time.

**Comparison to Existing Work**

The body of relevant existing research is fragmented and scattered across a number of disciplines. In addition, researchers are often concerned with only a particular group of users, a particular activity, or a particular domain. As a result, to develop a holistic and comprehensive understanding of interaction in the context of supporting complex cognitive activities, one must consult research from multiple disciplines, such as human-computer interaction, cognitive science, information visualization, visual analytics, information behavior, and learning technologies, and attempt to integrate such research into a coherent model. The manner in which EDIFICE-AP addresses this issue has been discussed in detail above and will not be repeated here.

Although there is a lack of general, comprehensive, and syncretic frameworks and models regarding human-information interaction in complex cognitive activities, researchers in the information visualization and visual analytics communities have been involved in developing one necessary component: interaction catalogs and taxonomies (e.g., Yi et al., 2007; Ward and Yang, 2003; Liu and Stasko, 2010; Gotz and Zhou, 2008; Heer and Shneiderman, 2012; Pike et al., 2009). Most of these, however, discuss only a small subset of possible actions and do not include actions...
identified in EDIFICE-AP such as animating, scoping, blending, assigning, and accelerating/decelerating, all of which are useful for complex cognitive activities mediated by different CASTs that are concerned with different information spaces. There are numerous information spaces and complex cognitive activities in which scoping, for instance, would be a desirable action, yet none of the existing work identifies and characterizes such an action and its utility. Another issue with existing work is that actions are sometimes presented without any characterization or examination of their utility. For instance, an action that has utility in many contexts (e.g., visual analytics, information visualization, decision support systems, and health informatics) is annotating. Much of the existing work does not identify, characterize, or describe the utility of annotating for performing complex cognitive activities. If annotating is identified (e.g., by Gotz and Zhou, 2008), its characterization is tied to a particular domain or activity, and therefore cannot inform a general framework concerned with interaction design.

In addition to these aforementioned issues, existing research often makes no clear distinction between different levels of interaction (i.e., activities, tasks, actions, events). As interaction is a complex phenomenon, such a distinction is crucial to establishing a consistent conceptualization and vocabulary for discussing interaction design. Yi et al. (2007), for example, identified 7 different interactions: select, explore, reconfigure, encode, abstract/elaborate, filter, and connect, each of which are characterized at different levels. For instance, select is a precise and low-level action. Explore, however, is a higher-level task that may actually involve lower-level actions such as selecting. In a similar fashion, Pike et al. (2009) identified explore as both a high-level task and an interaction; select as an interaction and selection as an interaction technique; and filter as both a low-level task and an interaction. In addition, they identified correlate and cluster as low-level tasks, and compare as a high-level task, without any characterization of these tasks or justification as to their ascribed levels. In another contribution, Liu and Stasko (2010) also did not make a clear distinction between different levels of interaction with information. For instance, they identified both explore and create as actions but then characterized them as activities. They also identified both save/load and explore as actions; however, save/load is a low-level and precise action, whereas explore is a high-level and imprecise task or activity.

Note: Vector art adapted from www.vectoropenstock.com under the Attribution Creative Commons 3.0 license

Figure 11: Different Actions Allow the User to Operate on Represented Information in Different Ways, and to Mentally ‘Reach Into’ an Information Space
Although existing work can be scrutinized to identify areas of needed improvement, the aforementioned researchers have faced the difficult task of characterizing and classifying a wide range of phenomena, and have made valuable inroads into bringing order and coherence to the vast landscape of interaction design. If we are to develop a science of interaction, however, much further research is required to characterize, categorize, and explicate the concept of interaction. Such a task requires a coherent integration of numerous issues regarding, among others, information, visual representations, cognition, perception, interaction, events, tasks, and activities. As demonstrated in the preceding pages, EDIFICE-AP represents a major attempt to provide a coherent, methodical, and comprehensive framework that contributes to such a research need.

**SUMMARY AND FUTURE WORK**

Cognitive activity support tools (CASTs) mediate and supplement human cognitive faculties to enable high-level activities such as making sense of phenomena, making decisions, solving problems, discovering knowledge in a large body of data, analyzing information, and learning. They do so by maintaining and processing digital information, displaying visual information representations (VRs) at their interface, and providing mechanisms through which users can interact with VRs. Due to their interactive nature, CASTs allow users to perform epistemic actions on VRs that facilitate mental information processing functions. This creates a strong coupling between the user and a CAST, and allows the tool to become an active participant in the user’s cognitive processes. The action choices offered by a CAST can extend the human mind, allowing users to reach into an information space to perform operations upon it, such as by bringing into view portions of the information that have not been encoded and displayed by the VRs at the interface level, by viewing information from different perspectives, or by reorganizing the information. When using CASTs to perform complex cognitive activities, it is through a sequence of epistemic actions that such activities emerge. Users engage in an interaction cycle in which they perceive VRs, interpret them and perform other mental operations, act upon them, perceive the reaction, and so on. This cycle continues until the user is satisfied with a task or until an overall activity is accomplished. Accordingly, interaction design for CASTs is concerned with what users can and should do with VRs, what actions should be made available to them, and what their subsequent reactions should be. In other words, interaction design is concerned with the discourse that takes place between users and VRs at the interface of a tool.
Researchers have recently recognised a need for developing a science of interaction that can guide the analysis and design of all kinds of tools that support complex cognitive activities. Although work has been done in this area, no existing frameworks are comprehensive enough to be applicable to all types of users, activities, tools, complex cognitive activities, and VRs. This paper attempts to address this need, and is part of a larger research effort to develop a comprehensive framework of human-information interaction with CASTs called EDIFICE (Epistemology and Design of human-InFormation Interaction in complex Cognitive activitiEs). Since this paper is largely concerned with the action part of EDIFICE, we have referred to it as EDIFICE-AP (where AP stands for Action Patterns). The focus of EDIFICE-AP is mainly on interaction at the level of individual actions and reactions, dealing mostly with pattern-based characterizations of actions and their utility in supporting complex cognitive activities.

Four major characteristics position EDIFICE-AP to address existing research needs. It is: 1) syncretic, unifying a number of previously disconnected ideas into a coherent theoretical model; 2) general, operating at a level of abstraction that is applicable to all kinds of technologies, activities, users, and VRs; 3) comprehensive, identifying patterns that cover an extensive range of actions; and, 4) generative, possessing the ability to motivate design creativity as well as to stimulate further theoretical and empirical research.

EDIFICE-AP can offer a number of benefits for researchers, designers, and evaluators of CASTs. First, it provides suggestions for design while still allowing for creativity, flexibility, and innovation at the implementation level. Since a pattern-based characterization allows EDIFICE-AP to be tool- and technology-independent, this flexibility also extends to tools and technologies. Consequently, it is resilient to technological change and extensible to future technologies such as tablets, interactive tabletops, motion sensing input devices, virtual reality and augmented reality environments, and interactive surface projection environments. A second benefit of EDIFICE-AP is that it provides a high-level support structure for communicating and thinking about interaction design in a systematic fashion. For instance, designers and evaluators may not be aware of certain action patterns and their utilities. Using EDIFICE-AP, they can methodically examine each action pattern to think about its utility and whether or not a CAST would be enriched by such an action. This allows for communicating and thinking about a wide range of action possibilities, and how they might benefit a user, in a systematic and consistent manner. A third benefit is that EDIFICE-AP is applicable to all activities. In CASTs, activities are emergent phenomena that result from the combining and chaining of numerous individual actions. Hence, whether designing or evaluating a tool for sense making, planning, learning, knowledge discovery, or problem solving, a subset of the epistemic actions in EDIFICE-AP can be used to support the activity, depending on the contextual and situational needs of the activity and its users. A fourth benefit is that EDIFICE-AP is applicable to all users. Since EDIFICE-AP is pattern-based, interactions can be implemented in such a way that it suits all age groups, levels of experience, capabilities, and backgrounds. Similarly, it is also applicable to both single-user and multi-user environments.

EDIFICE-AP provides opportunities for much future research. One important future line of research, for instance, can involve the investigation of the degree of information processing that should take place in the different spaces of the human-information interaction epistemic cycle for different types of CASTs and activities (see Figure 3). Currently, there is very little understanding of how processing load should be distributed among these spaces (see Parsons and Sedig, 2013b for a recent discussion of this issue). Another future line of research involves determining which action patterns complement one another in the performance of specific tasks and activities. Knowing which actions are complementary could allow for the creation of tools that support more coordinated and integrated tasks and activities. Another related line of research involves investigating the appropriate diversity and redundancy of actions for different tasks and activities. In other words, this line of research would be concerned with the number of actions that a CAST should offer, and whether users should be provided with multiple and diverse actions with which tasks and activities may be performed. Currently, we do not have a clear understanding of the implications of such considerations for interaction design. Future studies are required to develop a deep and structured understanding of these issues. Another possible area of research is in conducting empirical studies to develop a more detailed understanding of the utility of action patterns. In a general sense, each action pattern can support all kinds of activities; however, our knowledge of how and under what conditions each action pattern supports particular tasks and activities is still far from complete. Another possible area of future research involves categorizing interaction techniques according to the action pattern under which they fit. As there are hundreds of existing interaction techniques scattered across different disciplines, such a research effort could help to give more structure to the interaction design landscape. Additionally, such a categorization could add more of a prescriptive element to EDIFICE-AP and could provide a more robust palette of design rules and guidelines from which designers may make design decisions. Closely related to this is another line of action: that of devising new sets of different techniques and implementations under the same action pattern to compare, contrast and study their trade-offs. Furthermore, as EDIFICE-AP has presented many new action patterns, studies may be done to more accurately assess their relationships to particular tasks, users, tools, and complex cognitive activities.
ACKNOWLEDGEMENTS

This research has been supported financially by the Natural Sciences and Engineering Research Council of Canada. We would also like to express our gratitude to the reviewers of this paper for their constructive and helpful comments. This paper would not be in its current shape without their insightful suggestions.

REFERENCES

An, Y., J. Janssen, and E., Milios (2001) “Characterizing and Mining the Citation Graph of the Computer Science Literature,” Knowledge and Information Systems (6) 6, pp. 664-678.
Archim. www.archimy.com


Interaction Design for Complex Cognitive Activities with Visualizations


Gapminder. www.gapminder.org
GeoGebra. www.geogebra.org
Global Council Interlinkage. www.janwillemtulp.com/worldeconomicforum/


InfoZoom. www.infozoom.com


Interaction Design for Complex Cognitive Activities with Visualizations


Mendeley. www.mendeley.com


MindJet. www.mindjet.com


Panopticon. www.panopticon.com


Interaction Design for Complex Cognitive Activities with Visualizations

Sedig and Parsons


Sunaeon. www.sunaeon.com


VisANT (2011) http://visant.bu.edu


APPENDIX A. ADDITIONAL ACTION PATTERNS

The following provides a detailed characterization of the remaining 28 action patterns, along with their epistemic utilities, usage scenarios, and example CASTs in which they have been used. Although under each pattern we suggest certain activities that the action facilitates, because complex cognitive activities are emergent phenomena that result from the interaction of numerous actions and tasks, each action pattern can have utility in all activities. We have simply provided references to research in which an action has been shown to support some particular activity. Additionally, research with regard to the usage and effect of the different actions in the context of different activities is not evenly distributed. Despite the fact that we have tried to devote the same amount of space to each pattern, because some actions have been previously investigated more than others, we provide more references for some actions than others. However, EDIFICE-AP may encourage more systematic and balanced research to investigate the presented actions in the context of different information spaces, VRs, CASTs, and complex cognitive activities.

Unipolar Actions

Annotating

Characterization: Acting upon VRs to augment them with additional visual marks and coding schemes. This action creates a layer of personal meta-information on top of existing VRs. Here, “meta” signifies personalized editorial marks and commentary. Annotating does not inject information into the original information space.

Utility: Among others, this action facilitates learning, sense making, problem solving, and reasoning (Hwang and Shadiev, 2008; Schilit et al., 1998; Harris, 1990; Sedig et al., 2002; Wolfe and Neuwirth, 2001; Fast and Sedig, 2011; Marshall, 1997). In reasoning, problem solving, and sense making, augmenting VRs with additional information can facilitate reflective, inductive, and elaborative thinking (Ormrod, 1995; Kahney, 2003; Peper and Mayer, 1986). It supports and encourages users to build and strengthen connections between the represented information and their previous knowledge (Peper and Mayer, 1986; Foos et al., 1994). This in turn can facilitate critical thinking, which is fundamental to understanding and acquisition of further knowledge (Phelps and Wilensky, 1997; Schilit et al., 1998). Annotating can facilitate recall and reflection on past action, thereby supporting search for solutions to problems (Pimm, 1995; Preece et al., 2002; Sedig et al., 2002). Annotating can also help with text comprehension and reading. It can promote users’ meta-cognitive skills and aid understanding, memorization, and later retrieval (Slotte and Lonka, 1999; Hwang and Wang, 2004; Kiewra, 1989). Annotating, even in the form of simply underlining text, can be as effective as or equivalent to other learning strategies such as re-reading, answering periodic study questions, or summarizing (Anderson et al., 1984). The ability to spontaneously annotate information can facilitate sense making, even if users are not given a chance to review their annotations (Lonka et al., 1994; Lahtinen et al., 1997). Past research has shown that test performance and writing ability can improve as a result of annotating (Harris, 1990; Hynd et al., 1990; Strode, 1991). For instance, in a study by Liu (2006), it was noted that annotating while reading facilitated more critical thinking as reflected in students’ essays, and that the students with good annotating skills made exceptional progress in their writing.

Utility in CASTs: An example of a CAST that implements the annotating pattern is Tableau. Figure 13 shows a user analyzing the money dedicated to the economic development and welfare of developing countries. During the activity, she notices that Russia’s provision of aid is comparatively small. She then remembers reading a report that provided more information, and decides to annotate the VR with this personal meta-information to act as a reminder.

Note: Visualization by Stuart A. Thompson www.tableausoftware.com/public/gallery/economic-development-aid

Figure 13: An Implementation of the Annotating Pattern: Acting upon a VR to Add a Layer of Personal Meta-Information
Other usage scenarios: In a CAST containing VRs encoding a microbiology information space, given an image of a cell, learners may want to annotate it with some linguistic comments. In an adventure digital game, given a set of VRs encoding resources, players may want to attach a mark to a particular resource to be used later on. In a data analysis CAST containing a plot, users may want to attach labels or explanations to sub-systems of the plot (e.g., its data elements).

### Arranging

**Characterization:** Acting upon VRs to change their order. Some variations of arranging are moving, ordering, sorting, organizing, configuring, classifying, positioning, and ranking.

**Utility:** Among others, this action facilitates reasoning, problem solving, and sense making (Kirsh, 1995; Kastens et al., 2008; Peng, 2005; Yang et al., 2003; Siirtola, 1999; Piroli and Rao, 1996; Spence, 2007). Arranging allows users to explore and detect underlying patterns, dependencies, trends, correlations, and relationships in the represented information (Spence, 2007; Stolte et al., 2002; Siirtola, 1999). Arranging can also aid with the simplification of a representation space, trigger associations in the mind by presenting a fresh way of viewing the information, and aid with the organization of information (Kirsh, 1995; Kastens et al., 2008). For instance, often only by arranging rows and columns in a table, patterns and trends in information can emerge (Stolte et al., 2002). Arranging can affect a tool’s perceived clutter and structure (Peng, 2005). Arranging by bringing different entities of a VR closer together can be useful since the relationships among proximate elements are easier to detect than relationships among elements positioned far from each other (Yang et al., 2003).

**Utility in CASTs:** An example of a CAST that implements the arranging pattern is Table Lens, an information visualization tool for making sense of multivariate datasets (Rao and Card, 1994). In this CAST, users can interact with a table by arranging (e.g., sorting) its rows and columns, thereby discovering correlations between different variables (Piroli and Rao, 1996).

**Other usage scenarios:** In a pollution analysis tool, given a geographic map with icons representing different pollution-generating entities (e.g., factories, power plants, etc.), users may want to move a factory from one location to another to see its effect on the amount of pollution in a geographic region. In a CAST for making sense of financial data, given a tree diagram, users may want to re-arrange its sub-trees to experiment with different dependency relationships.

### Assigning

**Characterization:** Acting upon VRs to bind a feature or value to them (e.g., meaning, function, or behaviour). Some variants of this action pattern are designating and attributing.

**Utility:** Among others, this action facilitates sense making, investigating, reasoning, problem solving, and learning (MacKeracher, 2004; Kieran, 1989; Radford, 2002, 2006; Renkl, 1997). In mathematical problem solving and reasoning, assigning values to algebraic statements can facilitate reasoning about the generalization of patterns (Radford, 2006; Kieran, 1989). Assigning values to symbols in an algebraic statement can make it possible to see how higher-order meanings are made available for further affirmation (Radford, 2002). Assigning is also useful for forecasting activities—that is, predicting what will happen in the future (MacKeracher, 2004).

**Utility in CASTs:** An example of a CAST that implements the assigning pattern is NetLogo (Wilensky, 1999), a multi-agent programming language and modelling environment for simulating natural and social phenomena. Users can assign behaviours to hundreds or thousands of “agents” all operating independently. For example, while learning about the stability of predator-prey ecosystems such as those involving wolves and sheep, by assigning behaviours to the wolves or sheep, users can explore the connection between the micro-level behaviour of individual information elements and the macro-level patterns that emerge from the interaction of the collection of the individuals.

**Other usage scenarios:** In a CAST for exploring mathematical functions, given both a graph and a symbolic algebraic statement, learners may wish to assign different values to symbolic VRs to see what subsequent changes occur in the graph. In a problem solving CAST, given symbolic VRs representing agents that can carry out simple tasks, to solve problems and overcome obstacles users may want to dynamically select from a set of behaviors, functions, and capabilities and assign them to the symbolic entities to explore different problem solving strategies (e.g., as in the children’s game Lemmings in which the characters are assigned different functional properties).

### Blending

**Characterization:** Acting upon VRs to fuse them together such that they become one indivisible, single, new VR. Blending is different from composing in that once VRs are blended together they are not meant to be separated
again. The original VRs become indistinguishable from one another when blended together. Some variants of this action pattern are merging, fusing, and melding.

Utility: Little research has focused on the cognitive utility of this action pattern. However, it is likely that blending facilitates analysis, problem solving, and planning. In collaborative work environments, for example, multiple users often work with different copies of a VR. After working with these VRs, they can act upon the VRs to fuse them into one new VR, which can then be used to support group planning and decision making. In a similar fashion, a single user may be working with multiple copies of a VR and wish to blend them so that all of their features are incorporated into one new VR. Such is a common action in many productivity CASTs such as Microsoft Office.

Utility in CASTs: Hao et al. (2012) discussed an example of an implementation of blending in the context of visual analytics. They introduced an interaction technique called ‘motif merging’, which facilitates the exploration of frequently occurring patterns in time-series datasets. Thousands of items can be encoded within such large and complex datasets, resulting in visual clutter that hinders task performance. The authors’ technique allows the user to act upon a slider to set a threshold value, causing VRs within the specified value to fuse together. They suggest that this blending action reduces clutter and consequently facilitates the analysis activity.

Other usage scenarios: In a CAST for personal information management, a user may have a set of notes and ideas that are fragmented across different VRs within the tool. The user could select such VRs and act upon them to merge them together into one new VR. In a CAST for learning about chemical substances, a user may be working with a number of different VRs, each having different properties (e.g., solubility). As the user learns about each one, to support more complex higher-order thinking, the tool may encourage users to blend the VRs into one new VR that combines the properties of each. The new VR may then be used to enable and facilitate more complex tasks.

Cloning

Characterization: Acting upon VRs to create multiple identical copies of them. Some variant actions of cloning include copying, duplicating, multiplying, and replicating.

Utility: Among others, this action facilitates problem solving, investigating, and learning (Papadopoulos and Dagdilelis, 2008; Lamberty and Kolodner, 2002; Bauer and Wise, 2004; Clements et al., 2004; Jordan et al., 2006). In the context of problem solving and learning, Papadopoulos and Dagdilelis (2008) found that learners cloned VRs within a CAST to verify intermediate results or statements, such as verifying hypotheses about the number of shapes needed to fill up an area. In addition, while programming, being able to clone portions of code allows a programmer to build a high-level representation of the generic solution to a problem and apply it to multiple instances (Détienne and Bott, 2002). Cloning code allows programmers to reuse portions of software programs. In this way, cloning facilitates solving large problems more efficiently (Hoadley et al., 1996). When learning about visual patterns, cloning can allow the creation of complex designs to occur more quickly. This can facilitate deeper learning (Lamberty and Kolodner, 2002). Additionally, providing learners with facilities to be able to copy numbers and text can be useful when building writing skills and learning about numbers (Bauer and Wise, 2004; Jordan et al., 2006).

Utility in CASTs: An example of a CAST that implements the cloning pattern is Cytoscape (Shannon et al., 2003). Figure 14 shows a user exploring a network of yeast proteins during a knowledge discovery activity. Figure 14 (L) shows the user selecting a subset of the network VR to make a copy of it so that he can interact with the copied VR and not affect the original. Figure 14 (R) shows the result of the action, where the subset of the VR is copied into a new window that the user can then work with.

![Figure 14: An Implementation of the Cloning Pattern: Acting upon a VR of a Protein Network to Duplicate a Portion of it](image-url)
Other usage scenarios: In a virtual museum tool, users may wish to clone a digital artifact so they can explore it without affecting the original. In a drawing tool, while creating a collage, users may want to clone a picture to use it multiple times in the collage.

Comparing

Characterization: Acting upon VRs to determine their degree of similarity or difference, where similarity and difference can be in terms of proximity of or distance between value, meaning, geometry, topology, and/or other properties. Comparing can also be a higher-level task, but here it is only discussed as an action.

Utility: Among others, this action facilitates learning, experimenting, problem solving, analytical reasoning, and sense making (Star and Rittle-Johnson, 2009; Keller and Keller, 1993; Gentner et al., 2007; Ross, 1987; Kurtz et al., 2001; Darian, 2003; Davidson, 2003). Comparing is useful when two or more VRs need to be analyzed without explicit ordering and with no rank implied for them (Keller and Keller, 1993). Comparing is integral to many analytical processes of thinking, such as observing, modelling, and quantifying (Darian, 2003; Smith and Medin, 1981; Novick, 1990; Ross, 1987). In the context of learning, comparing is recommended as an important learning strategy (NCTM, 2000), and its benefits have been demonstrated in multiple case studies (Fraivillig et al., 1999; Huffred-Ackles et al., 2004; Lampert, 1990; Silver et al., 2005). Additionally, comparing may be a fundamental pathway to flexible, transferable knowledge (Rittle-Johnson and Star, 2007; Gentner et al., 2003).

Utility in CASTs: An example of a CAST that implements the comparing pattern is Multidatex (Wu et al., 2006), a tool for making sense of multivariate datasets. By providing several interactive, dynamically-linked VRs, this tool allows users to explore correlations in multivariate datasets. Two such VRs are a parallel coordinate plot and a network graph. Users can investigate how variables that affect air pollution, for example, are correlated, through comparison of the observations in the dataset. To compare and determine their degree of similarity, the user can select two or more observations from the parallel coordinate plot. Alternatively, the user can select a group of observations and request the tool to draw a network graph, where its nodes correspond to observations and its links represent the degree of similarity between the observations.

Other usage scenarios: In a visualization tool, given a topographic map, users may want to compare the length of different routes through the map. In a CAST for supporting software development, given multiple versions of software code, programmers may want to compare them to determine the changes made over time.

Drilling

Characterization: Acting upon VRs to bring out interior information that is not currently displayed. Drilling is usually not intended to alter VRs. Its main function is to penetrate into perceptually inaccessible, deep information items of the space and make them available for further investigation.

Utility: Among others, this action facilitates learning, reasoning, and investigating (Peng, 2005; Jonassen and Grabowski, 1993; Yi et al., 2007; Hannafin and Hooper, 1993; Buchel and Sedig, 2011). Drilling involves selective attention and encoding and is a fundamental action in many activities. It is a details-on-demand action (Plyshyn, 2003). Drilling can help make interesting objects easier to examine (Peng, 2005; Jonassen and Grabowski, 1993). Human attention plays a central role in most mental activities; however, it is a limited resource, and in complex fields of information users cannot attend to all information at once (Ormrod, 1995; Halpern, 2003). Consequently, by allowing users to focus attention on discrete items of information in detail, drilling can support convergent, narrow reasoning. Drilling usually involves shifting the focus of attention from broad scanning of an information space to narrow awareness of discrete elements in the space (Jonassen and Grabowski, 1993). In the context of learning, drilling allows users to process desired information more deeply, an important requirement for higher-order mental activities (Hannafin and Hooper, 1993).

Utility in CASTs: An example of a CAST that implements the drilling pattern is VICOLEX (Buchel and Sedig, 2011), a map-based visualization tool that acts as the front-end to a digital library. It supports users’ sense making of document collections. During a sense making activity, users can drill into VRs of geographical regions to get on-demand information about a collection and its different properties.

Other usage scenarios: In a geovisualization decision support tool, given a map with icons representing different localities, users may wish to drill an icon to have more information about a particular locality before deciding on a destination. In an exploration or decision-making tool, given a treemap visualization of different universities within different states, users may want to perform multi-level drilling (e.g., drill a state to get a listing of its universities and drill one of the universities to get its physical map).
Filtering

Characterization: Acting upon VRs to display a subset of their elements according to certain criteria. Filtering allows users to exclude some of the sub-systems of representations from view.

Utility: Among others, this action facilitates reasoning, problem solving, decision making, sense making, understanding, and learning (Stolte et al., 2002; Kastens et al., 2008; Strothotte, 1998; Stone et al., 1994; Marsh et al., 2004; Desimone and Duncan, 1995; Spence, 2007). As stated before, attention is a limited resource, and users often cannot attend to all information at once. Filtering allows users to notice trends and patterns in information, which is an integral part of analytical reasoning (Stolte et al., 2002). Filtering also allows users to have control over the degree of detail of a VR, sometimes removing non-essential details from the surface of the VR. Adjusting the level of detail is an important feature of the process of abstraction in the exploration of complex information spaces (Strothotte, 1998). Examining issues at a higher level of abstraction and with less detail, noise, and complexity is an essential aspect of most complex cognitive activities that involve generalization, categorization, and induction. For instance, in decision making, filtering can decrease perceptual and cognitive load by reducing the number of elements competing for attention in the visual field, allowing users to focus on a smaller subset of information (Strothotte, 1998; Desimone and Duncan, 1995).

Utility in CASTs: An example of a CAST that implements the filtering pattern is Global Council Interlinkage, a tool that supports exploration of survey data from Global Agenda Councils of the World Economic Forum. Users are initially presented with a VR that encodes hundreds of relationships, and is therefore too dense for most tasks. Users can act upon the VR to filter it such that only particular relationships are displayed, thus facilitating the performance of numerous tasks.

Other usage scenarios: In a personal information management tool (e.g., Microsoft Outlook), given a table (e.g., address book), users may wish to filter the elements based on a name or address criterion. In an educational tool, given a table (e.g., periodic table of elements), learners may wish to filter the table’s elements based on criteria such as group or year of discovery.

Measuring

Characterization: Acting upon VRs to quantify some of their items. Examples of measurable items of information include area, length, volume, mass, temperature, time, duration, speed, and distance from other items. Variations of measuring are calculating and quantifying.

Utility: Among others, this action facilitates decision making, learning, reasoning, and understanding, particularly in contexts that require quantification of information (Berka, 1983; Henshaw, 2006; Clements et al., 1997; Reynolds and Wheatley, 1996; Bishop, 1991). Measuring can be useful in the development, understanding, analysis, and solving of mathematical problems and ideas (Romberg and Kaput, 1999; Clements and Stephan, 2004; Miller, 1989). For instance, Norback and Love (1977) demonstrated that it is possible to solve the travelling salesman problem by measuring angles between cities on a 2D plane. Measuring can facilitate transitive reasoning5, which subsequently allows reasoning about units and iteration (Clements and Stephan, 2004; Piaget et al., 1960; Long and Kamii, 2001). Measuring can also facilitate comparative decision making (Bishop, 1991).

Utility in CASTs: An example of a CAST that implements the measuring pattern is GeoGebra. Figure 15 (L) shows how a user has drawn a shape and is about to measure the angle between three of its entities (i.e., points). Figure 15 (R) shows the result of the action.

Other usage scenarios: In a virtual museum tool, users may wish to measure the dimensions of ancient artifacts. In a digital library, users may wish to measure the number of words in a document. In a geovisualization tool, a user may wish to measure the distance between two locations.

Navigating

Characterization: Acting upon VRs to move on, through, and/or around them. Navigating can also describe higher-level tasks, in which case it involves subtasks such as identifying objects, moving, modelling, interpreting, and way finding. In this paper, however, navigating only concerns the action of moving. Navigating does not alter the representation on which it acts. Its variations in terms of utility are scanning and browsing.

Utility: Among others, this action facilitates learning, sense making, forming of concept maps, investigating, and knowledge discovery (Jul and Furnas, 1997; Dahlbäck, 1998; Spence, 1999; Jonassen and Wang, 1993; Lawless and Brown, 1997; Liang and Sedig, 2009; Sedig et al., 2003, 2005b). Navigating VRs helps with the formation of cognitive maps of an information space such as developing knowledge of its elements (or landmarks), relations (or routes), and structure (Liang and Sedig, 2009). As users navigate a VR, they acquire and modify their knowledge of
its structure, integrate knowledge of several routes into one network of routes, all being part of the process of creating an internal cognitive map of a VR. In learning, allowing learners to navigate and map their own paths of motion in an information space has positive outcomes (Jonassen and Wang, 1993; Fischer and Richards, 1995; Lawless and Brown, 1997). Navigation can also facilitate understanding semantic relationships between pieces of information, such as links between two elements in a VR. Additionally, navigating a VR that encodes social relationships can help users develop an understanding of such relationships.

Utility in CASTs: An example of a CAST that implements the navigating pattern is 3DLatticeViewer (Liang and Sedig, 2009), a tool that supports activities pertaining to 3D lattice structures. Figure 16 (from L to R) shows how a user can start from a point on the 3D lattice, which represents a chemical compound, and continuously move over its edges. By performing this action, the user can examine each connection in order to eventually discover how the chemical compound is structured.

Other usage scenarios: In modelling software, given a 3D model, users may want to navigate around the model to view it from different angles. In virtual museum software, users may want to navigate through a virtual room to see its artifacts. In a CAST visualizing the network setup of an organization, an administrator may want to navigate a VR representing the network to see if all of the links work properly.

Figure 15: An Implementation of the Measuring Pattern: Acting upon a VR to Quantify One of its Angles

Figure 16: An Implementation of the Navigating Pattern: Acting upon a VR to Move Across its Components

Searching

Characterization: Acting upon VRs to seek out the existence of or locate the position of specific items, relationships, or structures that satisfy certain criteria. Some variations of searching include seeking or querying.
Utility: Among others, this action facilitates sense making, understanding, investigating, and problem solving (Rowley and Hartley, 2008; Marchionini, 1997, 2006; Wolfe, 1998; Fast and Sedig, 2011). Searching is useful when users are aware that there is a knowledge gap, namely the idea that there is a distance between their contextual situation and the desired outcome (Marchionini, 1997; Kuhlthau, 1993; Dervin, 1992, 1997). When searching, users are actively attempting to answer questions or develop understanding around a particular question or idea. Users must generate the search and evaluate the results. Since it is impossible to fully process all of the stimuli in our visual field at one time (Tsotsos, 1990), searching supports the detection and selection of relevant information (Hannafin, et al., 1999). Searching allows users to actively replace or update mental models to make sense of a given situation (Klein et al., 2007; Marchionini, 1997; Dervin, 1977, 1983). Searching is often considered useful in problem solving (Marchionini, 1997; Gaslikova, 1999). Users may anticipate the content and possible sources of necessary information needed to solve a problem and execute a direct search, with which they can weigh the applicability of the results to their situation. Gaslikova (1999) notes that once a problem is structured and purposes are formulated, the strategy of information searching by means of exact retrieval requests seems to be the best strategy.

Utility in CASTs: An example of a CAST that implements the searching pattern is Health InfoScape, an e-health solutions tool. Figure 17 (L) shows the VR with which users are initially presented. Using this VR, it can be difficult to locate or discover the existence of specific items. Figure 17 (R) shows the result of the user searching for ‘neck pain’, in which case the relevant items are located and brought forward for viewing. After locating them, the user can perform further actions, such as drilling it for more information.

Figure 17: An Implementation of the Searching Pattern: Acting upon a VR to Seek Out the Existence of Items Related to ‘Neck Pain’

Other usage scenarios: In a complex and noisy network visualization tool, users may want to search a VR for smaller, constituent nodes representing information items. In a financial spreadsheet, users may want to search to locate a specific information item. In a CAST for investigating insurance fraud, given a large network diagram, an analyst may wish to search out the existence of a specific individual, transaction, or date.

Selecting

Characterization: Acting upon VRs to focus on or choose them. When applied to a set of VRs, selecting can perform a grouping function.

Utility: Among others, this action facilitates learning and investigating (Dalgano, 2004; Yi et al., 2005; Ward and Yang, 2004). Selecting can reduce cognitive demand (Brown, 1998). As we greatly depend on external information to reduce memory load, selecting a VR to set it apart or make it stand out can alleviate the cognitive load required to remember and/or keep track of it among other VRs. Selecting often precedes and is necessary for performing other actions within a CAST (Dalgarno, 2004). By selecting a VR and making it visually distinctive, users can easily keep track of it within a large amount of information, even when the VR is going through some changes (Yi et al., 2005). Selecting a set of VRs to group them together into a perceptual or cognitive unit has utility across a set of activities as well, such as reasoning, decision making, problem solving, learning, and investigating (Kastens et al., 2008; Henry and Fekete, 2006; Cooper, 1998; Kirsh, 1995; Lane et al., 2000; Munyofu et al., 2007). Group-based selection helps users deal with an aggregate entity rather than a larger number of individual objects (Newell and Simon, 1972; Cooper, 1998; Gobet, 1998). Dealing with aggregate rather than atomic items can alleviate memory load (Newell, 1994; Cowan, 2001). Grouping also facilitates cognitive processes involved in encoding, extracting, remembering, and understanding information (Gobet et al., 2001). Group-based selection supports rapid pattern recognition at perceptual and cognitive levels, a common strategy used by expert problem solvers (Feltovich et al., 2006; Roberston, 2001; Halpern, 2003). This action can also facilitate decision making and organization (Foster and Stefik,
1986). For instance, in many strategy exploration CASTs, allowing users to select objects into groups enables them to investigate group behaviour and decide how to organize and focus resources. Finally, when exploring VRs, selecting can provide various benefits: 1) it can encourage selective exploration and analysis; 2) it can invite conjectures of both similar and dissimilar attributes of elements; 3) it can facilitate focusing on a subset of elements of the VR; 4) it can support comparative reasoning of elements within groups and among groups; and 5) it allows performing operations on entire groups.

Utility in CASTs: An example of a CAST that implements the selecting pattern is Gapminder. This tool can be used, for example, to support sense making of relationships between income and life expectancy of multiple countries. When presented with a scatterplot, users can select a country or a group of countries in order to keep track of them. Doing so helps users to reason about their changes in relation to other countries as the tool displays temporal changes in the information space.

Other usage scenarios: In a chemistry visualization tool, given a set of symbols encoding different gas particles, users may want to select one of the symbols so as to keep track of it while adjusting parameters to do with pressure. In a sports analysis tool, given a video of a game, a coach may want to select one of its elements (i.e., a player) to monitor and analyze its movement. In a CAST for investigating trends in the stock market, users may wish to select a specific stock in order to keep track of it and see its temporal changes.

Sharing

Characterization: Acting upon VRs to make them accessible to other people or agents.

Utility: Among others, this action facilitates learning, planning, and decision making (Kirschner et al., 2009; Leidner and Fuller, 1997; Wu et al., 2009). Sharing has particular utility when a user is faced with a complex task, in that it allows the cognitive load required to perform the task to be distributed across the cognitive systems of multiple individuals. Kirschner et al. (2009) suggest that although such distribution requires information to eventually be reintegrated into the sharer’s mental space, when tasks require large amounts of cognitive processing, such costs are minimal compared to the gain achieved by the sharing. In contrast, they suggest that such costs may not be worthwhile in cases where tasks require only minimal cognitive load. Therefore, in terms of complex cognitive activities of a single user, the cognitive utility of sharing can be positively correlated with the complexity of the task at hand. In the context of performing collaborative complex cognitive activities, however, where all of the relevant information is not required to be integrated into one individual’s mental space, sharing seems to always have a positive effect. In the context of information systems and knowledge management, for instance, researchers have noted that organizational knowledge assets grow only at the rate at which individuals share information with others in a team or organization (Davenport and Prusak, 1998). In the context of complex decision making using map-based VRs, Wu et al. (2009) suggest that sharing can facilitate collaborative planning and decision making.

Utility in CASTs: An example of a CAST that implements the sharing pattern is Mendeley, a tool for organizing and managing research documents. With this tool, researchers can import papers that seem relevant to a task or activity to read at a later time. As a user discovers a paper of interest, she can act upon a VR of the paper to share it with her research team to support a collaborative activity.

Other usage scenarios: Using a visual analytics tool, after discovering an interesting trend, an epidemiologist may need to share a VR with a clinician or with a policy maker. In a CAST for organizational decision making activities, a user may be working with VRs to forecast revenue figures and wish to share it with a team of decision makers. With a CAST for genome analysis, a user may be working with VRs to explain genetic changes in the information space.

Transforming

Characterization: Acting upon VRs to change their geometric form. This action can alter the size, look, or orientation of VRs by rotating, scaling, magnifying, bending, folding, distortling, dilating, stretching, resizing, shrinking, and/or twisting them.

Utility: Among others, this action facilitates problem solving, learning, reasoning, sense making, understanding, exploring, and investigating (Wu and Shah, 2004; Peng et al., 2004; Elmqvist et al., 2010; Pinzger et al., 2008; Ward and Yang, 2004; Spence, 2007). While reasoning and understanding, transforming VRs can provide users with new perspectives on the information. Rotating, for instance, can facilitate sense making and reasoning about the structure of 3D objects that are presented in a 2D plane (Proffitt et al., 1992; Todd and Norman, 1991). In an experiment done by Wu et al. (2000), all students who were highly engaged in a CAST rotated diagrams of a molecular structure, which the researchers concluded was a crucial action in helping users to make sense of the novel diagrammatic structure. Transforming VRs in various ways can facilitate the exploration of large information spaces (Leung and Apperley, 1994; Elmqvist et al., 2010; Spence, 2007), especially when display area is limited. Transforming VRs can
be useful for allowing users to quickly explore their details without losing the larger context. Distorting, for instance, maintains visual and psychological continuity since users are able to see connections between the local detail and overall context (Card et al., 1999). A number of techniques have been developed in the information visualization community that can be used to transform VRs, such as fisheye, rubber sheet, hyperbolic, and x–y distortions (Card et al., 1999; Spence, 2007).

Utility in CASTs: An implementation of the transforming pattern is shown in Figure 18. This figure shows a distortion technique, Mélange (Elmqvist et al., 2010), that facilitates the exploration of large spaces; in this case a social network. As the VR encodes a large amount of information, it is impossible to investigate a portion of it closely while still retaining an overview of the whole VR. In a CAST that implements the transforming pattern, users can act upon the VR to alter its geometric form by folding it, enabling the investigation of portions of it in more detail without losing the larger context.

![Figure 18: An Implementation of the Transforming Pattern: Acting upon a VR to Fold it © 2010 IEEE.](image)

Other usage scenarios: In a forensic analysis tool, given an image of a fingerprint, an analyst may want to magnify the VR. In a mathematical visualization tool, given a plot in a Cartesian coordinate system, a learner may want to distort the coordinate system to observe its effect on the plot.

**Bipolar Actions**

**Accelerating/Decelerating**

**Characterization:** Acting upon dynamic VRs to increase the speed of movement of constituent components, or oppositely to decrease the speed.

**Utility:** Among others, this action facilitates learning, investigating, and sense making (Bétrancourt, 2005; Plass et al., 2009; Schwann and Riempp, 2004). A rigid pace of movement in dynamic VRs can put considerable cognitive load on users’ working memory (Hegarty, 2004), particularly if users cannot keep pace with the speed of movement of VRs (Hegarty et al., 2002). The pace of movement in a VR must be suitable for users so that they can make sense of a sequence of events (Tversky et al., 2002). Giving users control over the pacing of the presentation of dynamic information can improve both learning and comprehension (Plass et al., 2009; Tversky et al., 2002). A study conducted by Schwan and Riempp (2004) compared performance of subjects who could control (i.e., accelerate and decelerate) the speed of a video (i.e., dynamic VR) to those who could not, and results showed a significant decrease in time required to master the task by the former group. They also found that such actions were used more frequently when the information space was more complex. In addition, accelerating and decelerating a VR allowed the users to speed through or skip parts of a video that were perceived as easy to understand, and to focus on the more difficult parts (Schwan et al., 2000; Schwan and Riempp 2004).
Utility in CASTs: An example of a CAST that implements these two patterns is Netlogo (Wilensky, 1999), a multi-agent programming language and modelling environment for simulating natural and social phenomena. For example, while performing an expected-value analysis simulation—analyzing of the “value” of outcomes in probability experiments in terms of some utilitarian framework, such as money or points—Netlogo provides the ability for users to accelerate and decelerate dynamic VRs in order to support an overall analysis activity.

Other usage scenarios: In a physiology CAST, given a dynamic image (e.g., video) demonstrating the functioning of a human heart, users may want to accelerate or decelerate the contractions of the heart to support learning. In an educational tool, given an animation of nuclear decay, users may want to accelerate or decelerate the process of decay of atoms. In a CAST for studying astronomy, given an animation of the motion of the planets in a solar system, users may want to accelerate or decelerate the animation.

Animating/Freezing

Characterization: Acting upon dynamic VRs to generate movement in constituent components, or oppositely to stop the motion. Animating a VR causes a series of sequential VRs to appear in time, with each subsequent VR denoting a later temporal stage.

Utility: Among others, these actions facilitate problem solving, reasoning, learning, and sense making (Wong, 1994; Shah and Miyake, 2005; Rieber, 1990; Jones and Scaife, 2000; Sedig et al., 2003; Lawrence et al., 1994). In learning and sense making, animating VRs can impart more information about the dynamics of systems than could otherwise be obtained from equivalent static VRs. Animating can be used to illustrate complex structural, functional, and procedural relationships among objects and events (Jones and Scaife, 2000; Park and Gittleman, 1992). It helps users make sense of physical systems—e.g., lifted weights (Kaiser and Proffitt, 1987) and pendulums (Pittenger, 1985). When reasoning about spatial information, animating can help users perceive the shape and structure of 3D objects projected onto a 2D plane (Ullman, 1979). Additionally, animating can make spatial information and depth order salient, reduce spatial ambiguities within VRs, and help overcome users’ perceptual and cognitive biases that can be acquired from reasoning with static representations (Kaiser and Proffitt, 1987). For instance, Kaiser and Proffitt (1987) observed that many of the misconceptions that people have when asked to reason about physical systems do not occur when the same people are asked to make judgements about animated displays. Animating can also be very useful for making sense of information that has hidden and abstract meaning (Caraballo, 1985; Wong, 1994). For instance, it can be used to enhance understanding of dynamic physical or hidden biological processes such as the flow of blood in the human heart (Dwyer, 1994; Rieber, 1990; Rieber and Kini, 1991).

Utility in CASTs: An example of a CAST that implements these two patterns is Step, an educational tool that supports activities related to physics information spaces. Using this tool, users construct a simulation by repeatedly inserting information items (e.g., springs, particles, forces) into a VR. Users then act upon the VR to animate it, where the dynamic VR demonstrates the behaviour of gas particles under certain conditions, for instance. Users can animate and freeze the VR numerous times while performing a task or activity. This can aid users in understanding ideas that can be difficult to comprehend using static VRs.

Other usage scenarios: In a biology learning tool, given a VR of a cell, users may wish to both animate and freeze the process of cellular growth. In a tool for studying physical phenomena, given a VR of a wave, users may wish to animate it, and then freeze it part way through to make sense of the effect of wave interference. In a medical informatics tool, given a visualization of the spread of a particular disease, users may want to animate the visualization to observe the pattern of the spread of the disease.

Composing/Decomposing

Characterization: Acting upon VRs to assemble them and join them together to create a new, whole VR, or oppositely, break whole entities up into separate, constituent components. The goal of composing is often to build a larger VR than its constituent subcomponents; but the constituent elements or sub-systems need not be strongly associated. Variants of composing include assembling, building, and combining. Some variants of decomposing are fragmenting, disassembling, partitioning, and segmenting.

Utility: Among others, these actions facilitate problem solving, planning, learning, and reasoning (Gotz and Zhou, 2008; Abrahamson, 2006; Jane, 2006; Frederickson, 2003; Olive, 2000). Composing VRs can facilitate different forms of reasoning, such as analytical, deductive, syllogistic, and causal (Grossen and Carnine, 1990). In the context of syllogistic reasoning and problem solving, Grossen and Carnine (1990) found that children who were given the opportunity to compose diagrams achieved higher scores than those who worked with pre-drawn diagrams. The opposite action, decomposing, can lead to deeper thinking by allowing users to focus on both the aggregate as well as the individual items that compose a VR (Markman, 1979) and provides opportunities for discovering how the mechanisms in systems, such as tools, gadgets, and simple machines, work (Jane, 2006). While reasoning, decomposing can lead to perceptual and cognitive distinctions among discrete information items, and allows users to
analyze information in terms of smaller units (Gotz and Zhou, 2008; Rucker, 1987; Nickerson, 2004; Frederickson, 2003). In learning, decomposing can be critical to the development of increasingly complex concepts (Olive, 2000; Harel and Confrey, 1994; Lamon, 1999).

Utility in CASTs: An example of a CAST that implements these two patterns is SmartJigsaw (Ritter et al., 2002). With this tool, users are presented with numerous VRs of different components of the human foot. Depending on the task, users either act upon a VR to decompose it and create many separate VRs, or act upon VRs to bind them together into one whole VR (see Figure 19). Ritter et al. (2000; 2002) found that performing such actions with this CAST supported students’ learning and helped them to rehearse surgical procedures and dissection of cadavers.

![Figure 19: An Implementation of the Composing/Decomposing Patterns: Acting upon VRs to Bind them to Create a VR of the Human Foot or to Break them into Constituent Components](image)

Other usage scenarios: In a circuit exploration tool, given a set of VRs representing circuit components, users may want to assemble the components differently to study different types of circuits. In a digital library, given a VR of a document, users may want to decompose it to examine the different constituent components of the document.

Gathering/Discarding

Characterization: Acting upon VRs to place them into a collection, or oppositely to throw them away completely. Gathering is different from storing (discussed later) in that its purpose is short term. One can gather pieces of information into a pile to decide which ones to store to use at a later time. Gathering may be discussed in the context of higher-level tasks; here, however, we are referring to it specifically at the level of action. Discarding a VR has a permanent effect in that the VR gets completely expunged from the CAST and the user will not have any more access to it. Some variants of gathering are collecting and piling, and of discarding are scrapping, junking, annihilating, eliminating, and expunging.

Utility: Among others, these actions facilitate learning, problem solving, and planning (Hannafin et al., 1999; Price et al., 1998; Jones, 2004). In learning activities, gathering potentially important information can allow the information to be studied in closer detail or divided into subsets relevant to individual learning needs (Hannafin, et. al, 1999). In problem solving, for instance, collecting information into multiple folders in order to organize a problem space can be useful. Discarding supports all the above activities by getting rid of information that is of no interest.

Utility in CASTs: An example of a CAST that implements these two patterns is Hunter Gatherer (Schraefel et al., 2002), a tool that allows users to gather VRs from web pages into a collection for personal research and resource sharing. Figure 20 shows how users can select a VR from a web page, and by pressing a keyboard command, cause the VR to be gathered into a temporary collection. Users can also discard VRs that are no longer of use. According to the authors of the tool, users rarely gather VRs into such collections, in large part because of poor interaction design, and due to the fact that little focus has been given to the action of gathering itself (ibid.).
Other usage scenarios: In a digital library, given VRs of different research articles, users may want to gather together the VRs that seem interesting, and discard them after closer inspection later. In a visual analytics tool, users may wish to gather a number of VRs together in order to analyze their relationships.

Inserting/Removing

Characterization: Acting upon VRs to interject new VRs into them, or oppositely to get rid of unwanted or unnecessary portions. Inserting is different from annotating, as the added information is not meta-information; rather it is inserted in between the VRs’ elements and becomes an integral part of the existing VR. The difference between removing and discarding is that the removed information is not completely destroyed. Variations of inserting include embedding, enclosing, implanting, and adding.

Utility: Among others, these actions facilitate experimenting, reasoning, sense making, and problem solving (Avouris et al., 2003; Cohen and Gordon, 2008; Komis et al., 2002; Li et al., 2004; Sedig and Sumner, 2006). For instance Komis et al. (2002) investigated a problem solving tool that supports simultaneous development of diagrammatic VRs between dispersed collaborating partners. Either of the two participants can insert objects into a shared window to create multi-layered diagrams to solve problems collaboratively. Insertion of text in documents has been around for many years. Insertion may facilitate exploration and creative thinking, allowing users to pose what-if types of questions by interjecting information into VRs and observing the effect. Removing particular elements or regions of VRs allows users to focus on and work with relevant parts of information for further analysis. Removing can be a beneficial action for making sense of VRs that are composed of repetitive patterns, since only a portion of the pattern is needed for understanding the entire structure (Sedig and Sumner, 2006). Removal of image portions has been around for many years and is popular in film, television, publication, and photography (Li et al., 2004).

Utility in CASTs: An example of a CAST that implements these two patterns is ModellingSpace (Avouris et al., 2003), a tool that supports collaborative problem solving. With this tool, users take turns inserting VRs into and/or removing VRs from the representation space, enabling a gradual and collaborative problem solving process.

Other usage scenarios: In a chemistry simulation tool, given a VR of a container of water, a user can insert new salts into the VR and observe the changes in terms of solubility. In a visual analytics tool, an analyst may wish to insert hypothetical VRs (e.g., representing damage estimates for an insurance claim) into the representation space in order to see the effect on other VRs (e.g., of insurance premiums).

Linking/Unlinking

Characterization: Acting upon VRs to selectively establish a relationship or association between them, or oppositely
to dissociate them and disconnect their relationships. This action is different from composing in that the original VRs do not become combined into one whole; rather, they can remain as individual VRs that are linked to each other. Some variations of linking include connecting, relating, and associating.

**Utility:** Among others, these actions facilitate problem solving, planning, learning, sense making, understanding, and decision making (Uren et al., 2006; Yi et al., 2007; Wycoff, 1991; Peterson and Snyder, 1998; Dansereau, 2005; Foster and Stefik, 1986; Kaput, 1989). Linking VRs facilitates the establishment of connections between information items, which allows users to reason about relationships in the information space (Kaput, 1989; Spiro and Jehng, 1990; Godshalk et al., 2004). Thinking about possible connections or dissociations among information items is at the heart of reflective as well as divergent thinking (Zull, 2002; White and Gunstone, 1992). With the aid of linking, users can create different kinds of connections among VRs, such as causal, structural, semantic, temporal, and topological relationships (Markman, 1999; Jonassen, 2000). When a creative new idea is born, it usually consists of associations among information items in ways that may not have been previously considered (Massetti, 1996). Linking and unlinking allow users to experiment with different possible information scenarios. For instance, in the context of reasoning and understanding, linking and unlinking can facilitate critical thinking and allow users to see complex ideas in new ways, leading to deeper understanding of information spaces (Jonassen, 2000; Kaput, 1989).

**Utility in CASTs:** An example of a CAST that implements these two patterns is MindJet, a tool that supports planning, brainstorming, and task management. Figure 21 shows a user creating a concept map. In this instance, the user is explicitly linking VRs together by connecting them with lines. This type of linking allows users to make sense of how different concepts are related in a domain and to take a holistic approach to thinking about the concepts. The user can also unlink the concepts to explore new avenues of thought.

**Other usage scenarios:** In a financial analysis tool, given a map with icons representing different countries, users may want to explore different trade relation options by linking and unlinking the icons. In a data analysis tool, given multiple VRs, users may wish to link them together so that changes in one are propagated and reflected in others.

### Storing/Retrieving

**Characterization:** Acting upon VRs to put them aside for later use, or oppositely to bring VRs that have been put away into long-term storage back into usage in the working environment of a CAST. Storing is a more long-term action than gathering and allows users to save information for some anticipated future need. Some variations of storing include filing, saving, shelving, and archiving. The main variant of retrieving is restoring.

**Utility:** Among others, these actions facilitate problem solving and planning (Jones, 2004, 2008; Liu and Satsko, 2010; Gotz and Zhou, 2009; Anderson, 2000). Storing can be useful when users do not need information right away, do not have time to process it, or when users are interrupted and wish to maintain their current state to be resumed later (Jones, 2008; Czerwinski et al., 2004; Abrams et al., 1998). Both actions are generally useful for activities that take place over extended periods of time. When presenting users with VRs that represent large information spaces, giving them the option of storing and retrieving parts of the information is important (Barreau, 1995). In doing so, the cognitive burden of dealing with large amounts of information can be alleviated (Abrams et al., 1998; Norman, 1993). Additionally, users often have serendipitous encounters with VRs while performing tasks, where the information is not of immediate use, but has some perceived future benefit (Marshall and Jones, 2006). In such situations, storing and
retrieving can be helpful to them. In the context of planning, storing allows users to put aside information of interest with which they can plan events (Jones, 2008); for instance, storing information into specific organizational schemes such as folders can facilitate planning (Jones, 2004). In problem solving, often during its creative thinking and discovery component, users may reach a mental impasse, at which point it is often useful to save the current state of a problem and return to it after a delay; this delay, or incubation, can often facilitate the solution of the problem (Olton and Johnson, 1976; Smith, 1995; Simon, 1978; Anderson, 2000). Storing information to allow for incubation and future retrieval can contribute to insight experiences because the passage of time allows consciousness to fluctuate (Smith, 1995).

**Utility in CASTs:** An example of a CAST that implements these two patterns is Cytoscape (Shannon et al., 2003), a tool intended to support the exploration of complex networks (e.g., social networks, semantic networks, and molecular and genomic interaction networks). A user may be performing an analysis on a network, and after adjusting some of the properties of the network, wish to store it to be accessed in the future. The user can then retrieve the VR at some later time.

**Other usage scenarios:** In a digital library, while browsing, users may wish to bookmark or save an interesting VR for later access. In a visual analytics tool, when exploring an information space for the purpose of identifying financial fraud, users may notice a peculiar case and wish to archive it to examine it more closely later on.

---

\(^1\) For ease of reading and consistency, in this section we will refer to the designer as ‘he’ as to the user as ‘she’.

\(^2\) All figures in this section are screenshots of Panopticon, a CAST that supports visual analysis of numerous information spaces. Figures are used with kind permission from Panopticon (www.panopticon.com).

\(^3\) This is a form of reasoning involving the use of an item as a referent by which to compare objects and to deduce a relationship between them.
ABOUT THE AUTHORS

Kamran Sedig is an Associate Professor in the Department of Computer Science and the Faculty of Information and Media Studies at Western University, Canada. He holds a Ph.D. in Computer Science from the University of British Columbia. He has been doing research in the area of human-centered interactive visualizations since 1993. He is interested in the design of computer-based tools that help people perform information-intensive complex cognitive activities, such as sense making, decision making, data analysis, and learning. As such, his research and publications span a range of topics such as data and information visualization, visual analytics, human-information interaction design, information interface design, medical and health informatics, digital cognitive games, and cognitive and learning technologies. In the past few years, he has been working on the development of comprehensive frameworks that make the design and evaluation of visualizations and interactions more scientific.

Paul Parsons is a PhD candidate in computer science at Western University, Canada. His research explores technology-mediated human-information interaction to understand how interactive technologies can facilitate and enhance thinking and reasoning processes, with a particular focus on complex cognitive activities such as sense making, decision making, learning, and analytical reasoning. Application areas of his research include data and information visualization, visual analytics, information systems, medical and health informatics, and educational and cognitive technologies.

Copyright © 2013 by the Association for Information Systems. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Copyright for components of this work owned by others than the Association for Information Systems must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or fee. Request permission to publish from: AIS Administrative Office, P.O. Box 2712 Atlanta, GA, 30301-2712 Attn: Reprints or via e-mail from ais@aisnet.org.
### Editors-in-Chief

<table>
<thead>
<tr>
<th>Name</th>
<th>Institution, Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dennis Galletta</td>
<td>U. of Pittsburgh, USA</td>
</tr>
<tr>
<td>Ping Zhang</td>
<td>Syracuse U., USA</td>
</tr>
</tbody>
</table>

### Advisory Board

<table>
<thead>
<tr>
<th>Name</th>
<th>Institution, Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Izak Benbasat</td>
<td>U. of British Columbia, Canada</td>
</tr>
<tr>
<td>John M. Carroll</td>
<td>Penn State U., USA</td>
</tr>
<tr>
<td>Phillip Ein-Dor</td>
<td>Tel-Aviv U., Israel</td>
</tr>
<tr>
<td>Paul Gray</td>
<td>Claremont Graduate U., USA</td>
</tr>
<tr>
<td>Jenny Preece</td>
<td>U. of Maryland, USA</td>
</tr>
<tr>
<td>Gabriel Salvendy</td>
<td>Purdue U., USA, &amp; Tsinghua U., China</td>
</tr>
<tr>
<td>Ben Shneiderman</td>
<td>U. of Maryland, USA</td>
</tr>
<tr>
<td>Jane Webster</td>
<td>Queen’s U., Canada</td>
</tr>
<tr>
<td>K.K. Wei</td>
<td>City U. of Hong Kong, China</td>
</tr>
</tbody>
</table>

### Senior Editor Board

<table>
<thead>
<tr>
<th>Name</th>
<th>Institution, Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fred Davis</td>
<td>U. of Arkansas, USA</td>
</tr>
<tr>
<td>Traci Hess</td>
<td>U. of Massachusetts Amherst, USA</td>
</tr>
<tr>
<td>Shuk Ying (Susanna) Ho</td>
<td>Australian National U., Australia</td>
</tr>
<tr>
<td>Mohamed Khalifa</td>
<td>U. Wollongong in Dubai., UAE</td>
</tr>
<tr>
<td>Jinwoong Kim</td>
<td>Yonsei U., Korea</td>
</tr>
<tr>
<td>Anne Massey</td>
<td>Indiana U., USA</td>
</tr>
<tr>
<td>Fiona Fui-Hoon Nah</td>
<td>U. of Nebraska-Lincoln, USA</td>
</tr>
<tr>
<td>Lorne Ofman</td>
<td>Claremont Graduate U., USA</td>
</tr>
<tr>
<td>Kar Yan Tam</td>
<td>Hong Kong U. of Science &amp; Technology, China</td>
</tr>
<tr>
<td>Dov T’e’eni</td>
<td>Tel-Aviv U., Israel</td>
</tr>
<tr>
<td>Noam Tractinsky</td>
<td>Ben-Gurion U. of the Negev, Israel</td>
</tr>
<tr>
<td>Jewish</td>
<td></td>
</tr>
<tr>
<td>Mun Yi</td>
<td>Korea Advanced Ins. of Sci. &amp; Tech, Korea</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Editorial Board

<table>
<thead>
<tr>
<th>Name</th>
<th>Institution, Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miguel Aguirre-Ureeta</td>
<td>DePaul U., USA</td>
</tr>
<tr>
<td>Michel Avital</td>
<td>Copenhagen Business School, Denmark</td>
</tr>
<tr>
<td>Hock Chuan Chan</td>
<td>National U. of Singapore, Singapore</td>
</tr>
<tr>
<td>Christy M.K. Cheung</td>
<td>Hong Kong Baptist University, Canada</td>
</tr>
<tr>
<td>Matt Germonprez</td>
<td>U. of Wisconsin Eau Claire, USA</td>
</tr>
<tr>
<td>Jennifer Gerow</td>
<td>Virginia Military Institute, USA</td>
</tr>
<tr>
<td>Suparna Goswami</td>
<td>Technische U.München, Germany</td>
</tr>
<tr>
<td>Khaled Hassanein</td>
<td>McMaster U., Canada</td>
</tr>
<tr>
<td>Milena Head</td>
<td>McMaster U., Canada</td>
</tr>
<tr>
<td>Netta Iivari</td>
<td>Oulu U., Finland</td>
</tr>
<tr>
<td>Zhenhui Jack Jiang</td>
<td>National U. of Singapore, Singapore</td>
</tr>
<tr>
<td>Richard Johnson</td>
<td>SUNY at Albany, USA</td>
</tr>
<tr>
<td>Welling Ke</td>
<td>Clarkson U., USA</td>
</tr>
<tr>
<td>Sherrie Komiai</td>
<td>Memorial U. of Newfoundland, Canada</td>
</tr>
<tr>
<td>Na Li</td>
<td>Baker College, USA</td>
</tr>
<tr>
<td>Paul Benjamin Lowry</td>
<td>City U. of Hong Kong, China</td>
</tr>
<tr>
<td>Ji-Ye Mao</td>
<td>Renmin U., China</td>
</tr>
<tr>
<td>Scott McCoy</td>
<td>College of William and Mary, USA</td>
</tr>
<tr>
<td>Greg D. Moody</td>
<td>U. of Nevada, Las Vegas, USA</td>
</tr>
<tr>
<td>Robert F. Otondo</td>
<td>Mississippi State U., USA</td>
</tr>
<tr>
<td>Lingyun Qiu</td>
<td>Peking U., China</td>
</tr>
<tr>
<td>Sheizaf Rafaeli</td>
<td>U. of Haifa, Israel</td>
</tr>
<tr>
<td>Rene Riedl</td>
<td>Johannes Kepler U. Linz, Austria</td>
</tr>
<tr>
<td>Khawaja Saeed</td>
<td>Wichita State U., USA</td>
</tr>
<tr>
<td>Shu Schiller</td>
<td>Wright State U., USA</td>
</tr>
<tr>
<td>Hong Sheng</td>
<td>Missouri U. of Science and Technology, USA</td>
</tr>
<tr>
<td>Stefan Smolnik</td>
<td>European Business School, Germany</td>
</tr>
<tr>
<td>Jeff Stanton</td>
<td>Syracuse U., USA</td>
</tr>
<tr>
<td>Heshan Sun</td>
<td>U. of Arizona, USA</td>
</tr>
<tr>
<td>Jason Thatcher</td>
<td>Clemson U., USA</td>
</tr>
<tr>
<td>Horst Treibmaier</td>
<td>Purdue U., USA</td>
</tr>
<tr>
<td>Ozgur Turetken</td>
<td>Ryerson U., Canada</td>
</tr>
<tr>
<td>Fahri Yetim</td>
<td>U. of Siegen, Germany</td>
</tr>
<tr>
<td>Cheng Zhang</td>
<td>Fudan U., China</td>
</tr>
<tr>
<td>Meiyun Zuo</td>
<td>Renmin U., China</td>
</tr>
</tbody>
</table>

### Managing Editors

<table>
<thead>
<tr>
<th>Name</th>
<th>Institution, Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jian Tang</td>
<td>Syracuse U., USA</td>
</tr>
</tbody>
</table>

### SIGHCI Chairs

<table>
<thead>
<tr>
<th>Year</th>
<th>Chair</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001-2004</td>
<td>Ping Zhang</td>
</tr>
<tr>
<td>2004-2005</td>
<td>Fiona Fui-Hoon Nah</td>
</tr>
<tr>
<td>2005-2006</td>
<td>Scott McCoy</td>
</tr>
<tr>
<td>2006-2007</td>
<td>Traci Hess</td>
</tr>
<tr>
<td>2007-2008</td>
<td>Weiyin Hong</td>
</tr>
<tr>
<td>2008-2009</td>
<td>Eleanor Loiacono</td>
</tr>
<tr>
<td>2009-2010</td>
<td>Khawaja Saeed</td>
</tr>
<tr>
<td>2010-2011</td>
<td>Dezhi Wu</td>
</tr>
<tr>
<td>2011-2012</td>
<td>Dianne Cyr</td>
</tr>
<tr>
<td>2012-2013</td>
<td>Soussan Djamalsbi</td>
</tr>
<tr>
<td>2013-2014</td>
<td>Na Li</td>
</tr>
</tbody>
</table>