KNOWLEDGE TRADING
Computational Support for Individual and Collaborative Sense-Making Activities

by Paul Erich Keel

Master of Science in Architecture, February 1997
Massachusetts Institute of Technology

Diplomierter Architekt HTL, June 1992
Zentralschweizerisches Technikum Luzern, Switzerland

Submitted to the
Department of Architecture

in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Architecture: Design and Computation

at the
Massachusetts Institute of Technology

September 2004

Signature of the Author
Paul Keel
Department of Architecture
August 6th, 2004

Certified by
William L. Porter
Norman B. and Muriel Leventhal Professor of Architecture and Planning
Thesis Supervisor

Accepted by
Stanford Anderson
Professor of History and Architecture
Chairman, Department Committee on Graduate Students

DISSERTATION COMMITTEE

William L. Porter
Norman B. and Muriel Leventhal Professor of Architecture and Planning, MIT

William J. Mitchell
Professor of Architecture and Media Arts and Sciences, MIT

George Stiny
Professor of Design and Computation, MIT

Patrick H. Winston
Ford Professor of Artificial Intelligence and Computer Science, MIT

READERS

Edith Ackermann
Professor of Developmental Psychology, Université de Provence, Aix, Marseille

Jeffrey Huang
Associate Professor of Architecture, Graduate School of Design, Harvard University

SUPPORT

Agent-Based Intelliegent Reactive Environments Research Group, MIT

Michael Fox
Professor of Architecture, Pasadena College of Art and Design
Lecturer of Architecture, Southern California Institute of Architecture

Chiai Takahashi
Senior Graphic Designer, Ascent Technology Inc.

KNOWLEDGE TRADING

Computational Support for Individual and Collaborative Sense-Making Activities

by Paul Erich Keel

Submitted to the Department of Architecture on August 6th, 2004 in partial fulfillment of the requirement for the degree of Doctor of Philosophy in Architecture: Design and Computation at the Massachusetts Institute of Technology.

Abstract

This dissertation explores the potential for computational systems to analyze and support individual and collaborative human sense-making activities. In this context human sense-making refers to the act of mentally and physically relating pieces of information so as to develop an understanding of a particular situation. Human sense-making activities such as brainstorming, decision-making, and problem solving sessions often produce a lot of data such as notes, sketches, and documents. The participants of sense-making activities usually develop a good understanding of the relations among these individual data items. These relations define the context. Because the relations remain within the minds of the participants they are neither accessible to outsiders and computational systems nor can they be recorded or backed up. This dissertation outlines a first set of computational mechanisms that construct relations from the spatial arrangement, use, and storage of data items. A second set of computational mechanisms takes advantage of these relations by helping users to keep track of, search for, exchange, arrange, and visualize data items. The computational mechanisms are both adaptive and evocative, meaning that the computational mechanisms dynamically adapt to users and changing circumstances while also trying to influence the human sense-making process.

Contributions

1. Demonstration that computer systems can discover probable relations among data items from their spatial arrangement and use by users.

This work identifies and analyzes various human mental processes involved in the determination of possible relations among data items such as documents on a work desk or files in a computer system. A computational equivalent is proposed for every mental process outlined.

2. Demonstration that computer systems can use the discovered relations among data items to help users search for relevant information, prioritize the data exchange between collaborating users, and visualize data in various ways

This investigation looks at how a human's increasing knowledge about a problem space is influential in the subsequent accumulation of new data. The findings are converted into computational equivalents that can support individual and collaborative sense-making processes.

Thesis Supervisor

William L. Porter
Norman B. and Muriel Leventhal Professor of Architecture and Planning

RESEARCH TEAM

2000 - 2004 Patrick H. Winston, Project Investigator
Ford Professor of Artificial Intelligence and Computer Science, MIT

2000 - 2004 William L. Porter, Project Investigator
Norman B. and Muriel Leventhal Professor of Architecture and Planning, MIT

2000 - 2004 Edith Ackermann, Consultant
Professor of Developmental Psychology, Université de Provence, Aix, Marseille

2000 - 2004 Jeffrey Huang, Consultant
Associate Professor of Architecture, Graduate School of Design, Harvard University

2000 - 2004 Paul Keel, Project Supervisor
Graduate Student in Architecture and Urban Planning, MIT

2002 - 2004 Mike Kahan
Graduate Student in Electrical Engineering and Computer Science, MIT

2002 - 2004 Yao Li
Graduate Student in Electrical Engineering and Computer Science, MIT

2003 - 2004 John Bevilacque
Graduate Student in Electrical Engineering and Computer Science, MIT

2003 - 2004 Mathew Sither
Undergraduate Student in Electrical Engineering and Computer Science, MIT

2003 Benjamin Wang
Undergraduate Student in Electrical Engineering and Computer Science, MIT

2002 Raudel Rodriguez
Undergraduate Student in Electrical Engineering and Computer Science, MIT

2002 Akshay Patil
Undergraduate Student in Electrical Engineering and Computer Science, MIT

SPONSOR

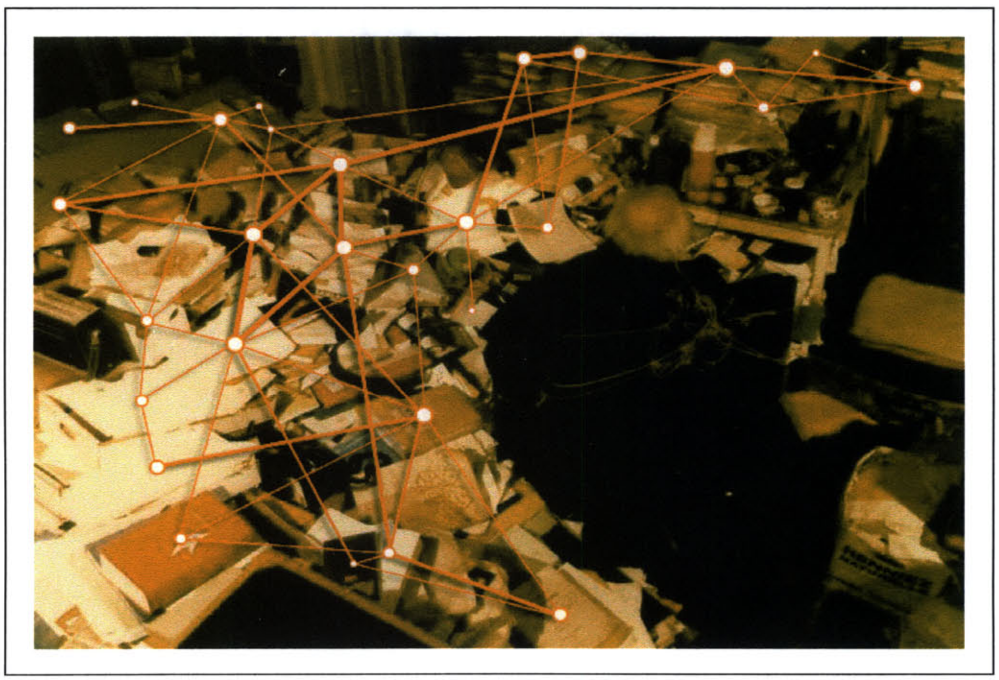
This work was supported in part by

2001 - 2004 ONR (Office of Naval Research)
Cognitive, Neural and Biomolecular Science and Technology Division
Grant N000140210132

TABLE OF CONTENTS:

INTRODUCTION

This work explores the potential for computational systems to analyze and support individual and collective human sense-making activities. In this context, sense-making refers to the act of mentally and physically relating pieces of information so as to develop an understanding of a particular situation. Human sense-making activities such as brainstorming, decision-making, problem solving, and designing often produce great amounts of information such as notes, sketches, and documents. The participants of sense-making activities are supposed to develop an understanding of the relations among the individual information items. Because these relations often evolve unconsciously and remain within the minds of individuals they are neither accessible to outsiders and computational systems nor can they be recorded or backed up. This work introduces a set of computational mechanisms that infer explicit and implicit relations from the organization and use of information. These relations primarily reflect the spatial arrangement and collaborative use (syntax) rather than the contents (semantics) of information. A second set of computational mechanisms utilizes these relations as a means to help people administer information.



This work investigates various human cognitive concepts involved in inferring and utilizing relations among information items and introduces one corresponding computational mechanism for every cognitive concept examined. The approximated associations between the computational mechanisms and cognitive concepts are valuable to analyze, understand, and computationally support human sense-making activities. The computational mechanisms are divided into two groups: Interpretation and Transformation Algorithms. The former establish probable relations among information items based on the organization, history, and collaborative use of information while the latter utilize these relations as a basis for collecting relevant information, prioritizing information exchanged among multiple collaborating users, and arranging information in various familiar, inspiring, and diagnostic styles.

This work resulted in the development of a software application aimed at supporting individual and collective sense-making activities. The software application introduces a flexible computational framework that allows humans to quickly view, collect, organize, and communicate large amounts of information as well as to facilitate the collaboration among humans with different levels of involvement in large, distributed and decentralized teams across organizational boundaries. The computer application is divided into five Modules. Each Module is focused on supporting one particular group of sense-making activities: The Workspace Module helps users to collect, organize, and comprehend information. The News Module helps users to monitor for additions and modifications to information sources. The Database Module combines and structures information contributed by collaborating users. The Exchange Module prioritizes information exchanged among collaborating users. The Visualization Module analyzes and visualizes information.

Note: Section A on EWall introduces the concept and functionality of the software application thus providing the basis for understanding the underlying computational mechanisms. Section B on Algorithms introduces the various observations of human cognitive processes and their subsequent translation into computational mechanisms.

Note: Some of the functionality introduced in Section A and B has not yet been implemented. Detailed information about the implementation status is provided at the end of Section A and B.

Note: The software application outlined in Section A and the computational mechanisms outlined in Section B were conceived by Paul Keel and implemented by Michael Kahan, Yao Li, Akshay Patil, Raudel Rodriguez, Mathew Sither, Benjamin Wang, and Patrick Winston. The development of the software application and computational mechanisms progressed under the consultation of Edith Ackerman, Jeffrey Huang, William Porter, and Patrick Winston. (see Research Team)

SECTION A EWALL

EWall is an acronym for Electronic Card Wall and used for both the name of this research project as well as the name of the software application developed within the framework of this research project. The objective of the EWall Project is to investigate human sense-making activities with a focus on social interactions that improve the ways in which humans comprehend and share information. The objective of the EWall Application is the development of a flexible computational framework for the support of individual and collective human sense-making activities (Computer Supported Sense-Making). The EWall Application does not present a comprehensive solution for the support of all sense-making activities but offers a series of independent mechanisms for a variety of possible applications.

The EWall project originated in 1996 from within the domain of architectural design and initially focused on the study of individual and collaborative sense-making activities as well as the physical environments and tools suitable for such activities. The goal was to study existing activities and tools as well as to develop new computational means for the support of people working on explorative and creative tasks. The scope of this project widened quickly to draw upon research in psychology, cognitive science, artificial intelligence, organizational management, and information technology. In 2001, the research conclusions translated into specific concepts for the development of the EWall Application. Significant portions of the EWall Application were realized over the subsequent years with a specific focus on its use for decision-making within command and control environments. As of 2004, several components of the EWall Application were made available for evaluation and testing to NAVY related organizations such as NAVAIR, SPAWAR, NPS, and NSW as well as commercial organizations such as ARUP and Saab Aerospace [1].

Design Principles

The design of the EWall Application is governed by five principles that address issues in human computer interaction, information management, and software design:

1. Adaptation of existing rather than imposition of new work processes

Humans develop unique sense-making processes and dynamically adjust these processes to changing circumstances. While propositions for standardized sense-making processes may help humans to deal with abstract and time-constrained tasks, standardized sense-making processes are often counterproductive for the execution of explorative and creative tasks. This is because the creation of new and unique ideas, views, and solutions primarily emerge through the individuals' unique modes of working and thinking as well as their distinct backgrounds, expertise, motivations, interests, and foci. The first design principle of the EWall Application is not to propose a particular work process and way of using the system but rather a flexible and adaptive computational framework capable of supporting a diverse range of applications, users and circumstances.

2. Interpretation of user activities through observation rather than user responses

Sense-making often happens in the minds of humans thus making it difficult for computational tools to directly support such activities. For example, a human might detect some idiosyncratically relevant similarities between two text documents. This observation is influential to the human's subsequent investigation yet inaccessible to a computational system. This problem is commonly dealt with by requiring users to be explicit about their findings and conclusions. For example, a human could be asked to report all discovered relations among documents as a basis for electronic processing. The problem is that such procedures interrupt and defer attention from human sense-making activities. The second design principle of the EWall Application deals with this issue through inferring from computational observations of user activities rather than distracting users with questions about their findings and conclusions.

3. Focus on inspiring rather than directing users

Sense-making is an ongoing process that combines existing knowledge with new information. Consequently, sense-making activities depend on the availability of relevant information. The relevance, availability, and applicability of information for specific sense-making tasks are often difficult to determine. For example, information retrieval software such as web search engines might return search results that accurately match a specific query. However, sense-making tasks not only benefit from information that precisely fits a specific domain but also from information that inspires different perspectives, questions previous considerations, and encourages explorations into similar or tangentially related domains. The third design principle of the EWall Application is to provide users with information that may inspire alternative views and to avoid singularly directional sense-making processes. In other words, the EWall Application does not want to internalize intelligence but to engage, utilize and foster the intelligence of humans.

4. Focus on the context rather than the content of information

Sense-making involves the accumulation of content as well as the adaptation and formulation of context. Content refers to data, information, and knowledge while context refers to the relations among content items. Computer applications are more commonly used for the management of content while humans seem more successful with the formulation and analysis of context. Computational possibilities for the formulation of context are often limited to the comparison of words and file information contained in electronic documents. For example, a computational system might relate two text documents because both documents contain the same uncommonly used word. The fourth design principle of the EWall Application is to formulate context based on an analysis of the history, organization and collaborative use of information rather than only a comparison of information contents. The computational mechanisms introduced by the EWall Application are designed to explore alternative and human-like ways of formulating context that complement rather than substitute existing methods.

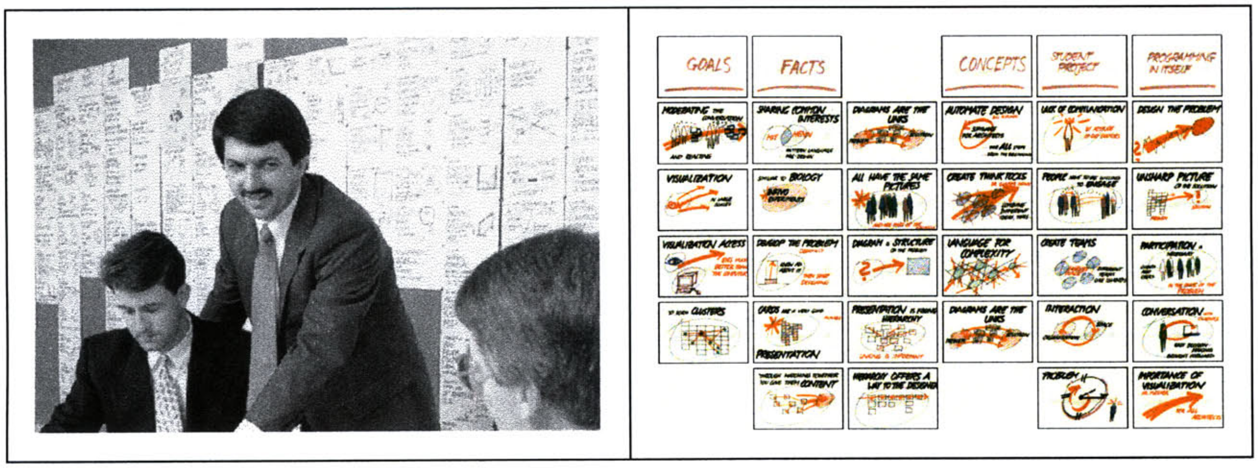
5. Focus on modular rather than integrated solutions

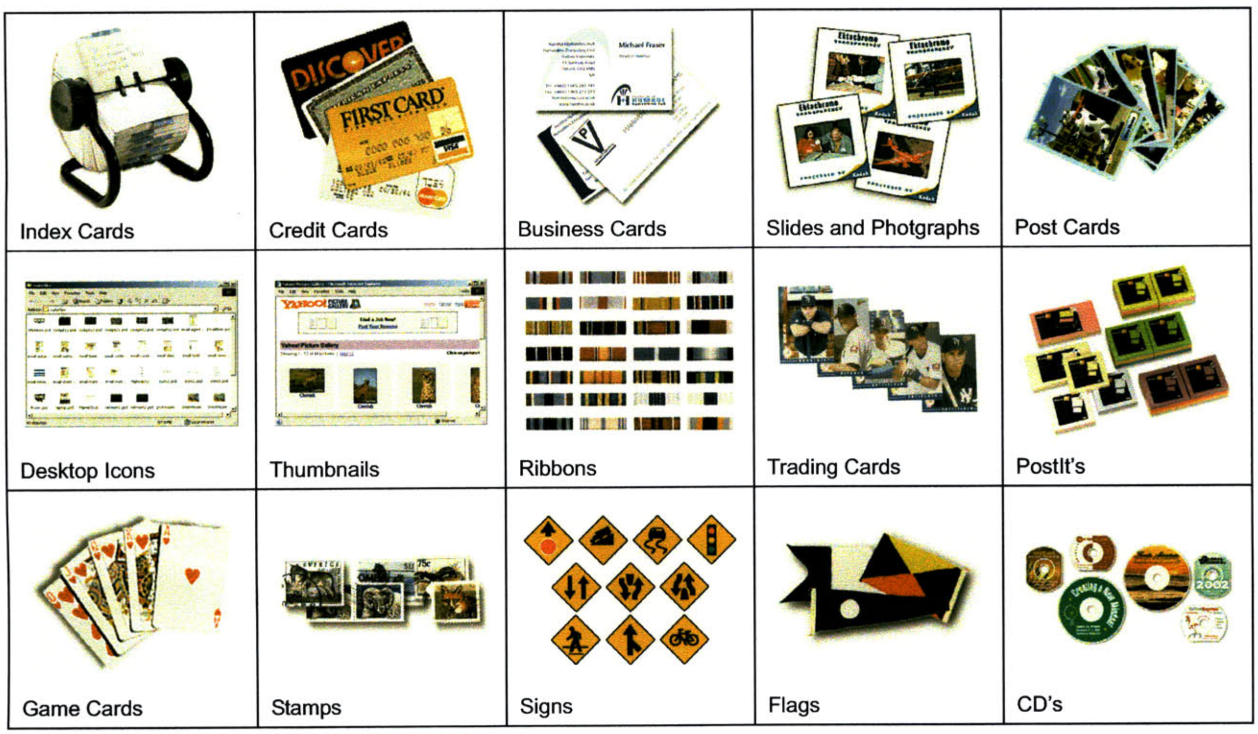
Humans choose among a variety of tools to support their sense-making activities. The tools help humans to externalize knowledge, visualize information, and search for relevant data. Typical tools include whiteboards and sheets of paper for drawing and writing, pin boards and tabletops for spatially arranging documents, and web search engines for exploring relevant data on the Internet. The choice and combination of tools depends on the user's experience and preference, the task, and the current circumstances. The ability to combine and choose from a variety of tools is essential for humans to deal creatively and effectively with unique sense-making tasks. The fifth design principle of EWall is to provide users with a highly modular application that can be combined and customized for a variety of different users, settings, and applications. The modularity of the EWall Application not only allows for more flexibility but also ensures user control over most parts of the software by allowing users to add and remove individual software components.

Problem Seeking

The EWall Project builds on William Pena's Problem Seeking methodology, a research conducted at CRS [2] and first published in 1977 [3]. The methodology was conceived to support meetings with large numbers of participants in shared physical locations and exclusive of computer technologies. The methodology introduces processes and techniques for the collaborative recording and organization of issues conceived during meetings. The key concept is to capture comments, suggestions and ideas on small paper cards. The cards are pinned up on walls and organized under predefined categories such as Goals, Facts, Concepts, Needs, and Problems (see Illustration 1a). Participants continuously add, compare, discuss and rearrange cards in order to develop a shared understanding of their various opinions and suggestions. A card usually contains a keyword, a graphical icon and some explanatory text. While the keywords are often sufficient to remind people of a particular issue, the graphical icons help people to memorize and locate cards. The creation of cards may be seen as a means to externalize the knowledge of individuals and to allow this knowledge to enter the domain of discourse of all participants. Since its introduction, Pena's methodology has become popular for meetings of various kinds. These meetings are not always conducted according to Pena's specifications but have since produced multiple methodological variations.

The advantages of Pena's methodology include the fast accessibility to large amounts of information through the physical representation, graphical enhancement, and clear categorization of individual information items. The contents developed during meetings remain visually accessible to everyone thus allowing participants to keep track of previously discussed issues, to switch more easily between subjects and to explore relations among individual contributions. This evolving information space also provides a basis for participants to establish a group identity, to sustain mutual awareness, and to develop a shared understanding. The accumulation and combination of individual contributions may be viewed as a shared memory or discussion record whose contents and relations among content items continuously change as determined by the collaborative effort of all participants.





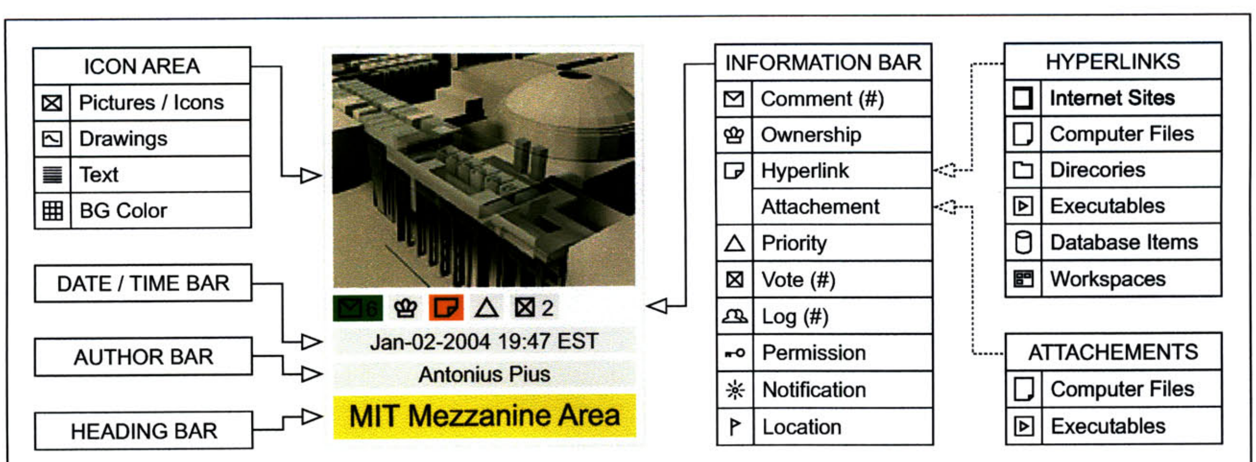


Illustration 1a: Problem Seeking Session (HOK) and Card Wall (HENN); 1b: Information Objects; 1c: EWall Card Components

The disadvantages of Pena's methodology lie in the static nature of card arrangements, the fast accumulation of cards, and the various efforts involved in conducting professional Problem Seeking sessions. Large card arrangements are difficult to rearrange and restore. Moreover, there is a limit to the amount of cards that humans can visually and mentally relate. Card arrangements also do not effectively reflect the dynamically changing relationships among cards. Furthermore, the creation and the grouping of cards require skilled people not only capable of understanding but also of evaluating, abstracting and graphically representing the issues discussed during meetings. As a consequence, these types of meetings are often rendered by professional services that employ and train people specifically for such assignments. The dependency on skilled people prevents the spontaneous application of Pena's methodology.

Essential conclusions from the analysis and subsequent studies of Pena's work reflect on the ways in which people convert, standardize, abstract, associate, and relate data, information and knowledge. These particular activities are not only encouraged through the creation and use of cards but are also present in the creation and use of several everyday objects such as index, trading and game cards (Illustration 1b).

1. Converting

Humans like to think of data, information and knowledge as objects that they collect, compare, and organize. The conversion of data and information (as well as the externalization of knowledge) into virtual and physical objects accommodates this way of thinking. Dealing with virtual and physical objects such as files on a computer desktop or documents on a table enables humans to engage their motor abilities, vision and touch senses. The combined use of mind, body, and sensory functions effectively increases a human's ability to deal with complex sense-making tasks. For example, a professor memorizing the names of his students usually engages his visual senses by associating the names and faces of students during teaching hours. The professor might also choose to create small index cards that display the pictures and names of students thus allowing the professor to easily remove the cards of students whose names he already memorized and to spatially arrange cards in ways that best support his way of memorizing.

2. Standardizing

A standardized card size and layout is convenient to collect, compare, and organize cards. The benefits of standardized objects are present in various everyday objects. For example, index cards, credit cards, business cards, slides, photographs and postcards are usually of equal size and layout so they can conveniently be stored, accessed, and processed. One of the disadvantages of standardized objects is their dependency on predefined templates. These templates have to account for a wide range of eventualities as later modifications of objects modeled after a particular template can become very complicated and time intensive. For example, a professor might create index cards for each of his students indicating their names, countries of origin, and ages. Later additions, such as for example the inclusion of the students GPA's, would require a modification to the card template as well as the subsequent adjustment of all previously created cards. If on the other hand the professor had conceived a template that accounts for all eventualities then most of the card space would be occupied by rarely used information.

3. Abstracting

The abstract representation of data, information and knowledge with cards engages a human's visual cognition in ways that increases information access time and allows for the processing of large amounts of information. Furthermore, the process of creating cards requires users to circumscribe the contents associated with cards in a visually and mentally fast accessible and easily comprehensible format thus encouraging a more careful analysis and understanding of the contents associated with cards. The card template suggested by Pena consumes little space and offers a good balance between abstract visual and textual reminders. It allows users to easily memorize and recall the contents associated with cards as well as to quickly locate and compare cards. The concept and use of abstract visual and textual reminders is also present in various everyday objects. For example, desktop icons and thumbnail views allow users to easily locate and organize computer files. Military ribbons use abstract visual representations to provide service, mission and award specific information on a small clothing area. Traffic signs depend on abstract visual representations that are easy to spot and understand by pedestrians and car drivers.

4. Associating

A card usually does not contain information per se but only serves as a reminder for the presence of a particular piece of data, information or knowledge. The separation between cards and content associated with cards allows for the compact visualization and organization of large amounts of content. A card may be viewed as a meaningfully labeled hyperlink to a piece of content available in a remote location. This particular function is embodied in several everyday objects such as Post-it's and trading cards. Post-it's are commonly used for taking notes and for labeling physical objects. Trading cards usually reference people and physical objects. Even though the printing costs, the layout, and the amount of information contained on a trading card do not differ significantly, the trading card values vary tremendously. Typically the value of a trading card increases if the trading card closes a gap in a sequence of trading cards or if the trading card represents a popular person or object. In other words, the trading card value emerges through its relation with other trading cards and through its association with particular instances of human knowledge.

5. Relating

Pena's Problem Seeking methodology engages users in a process of arranging and rearranging cards. Users benefit from this process by developing a good understanding of the card contents and the relations among cards (context). The use of game cards displays interesting parallels with Pena's card arrangements. Players arrange game cards in an attempt to explore and visualize groups and sequences of game cards. The distribution and arrangement of game cards among players creates a meaningful context that was not present in the previously shuffled deck of game cards. In other words, the meaning contained in an arrangement of game cards exceeds the meaning contained in the combined contents of all game cards. However, the game card contents as well as the subsequently created context are often only meaningful to those players that know the game development and that understand the rules of the game. The same is true for most arrangements of objects meaning that different people often interpret arrangements of objects differently. Complementing arrangements with explicit hints about their organizational structures may reduce the number of possible interpretations. For example, newspapers provide hints about their organizational structures through the positioning of text blocks and headings as well as the use of distinct font styles and font sizes. The absence of such hints may render essential contextual information inaccessible to outsiders. This problem is particularly noticeable in Pena's Problem Seeking methodology where often only the authors of a card arrangement understand its hidden meaning. Another observation regarding Pena's Problem Seeking methodology refers to the use of card arrangements during collaborative sense-making tasks. People of different backgrounds, interests and foci have their unique ways of relating information. The collaborative development of card arrangements can help people to determine intersecting views as well as to develop a shared understanding of a particular information space. Imagine for example two people organizing a set of stamps. One person might be more mathematically focused thus preferring an organization by stamp sizes and values. The other person might be more visually oriented thus preferring an organization by shapes and colors. Through the collaborative effort the two parties learn about each other's views and might even conclude their efforts with a solution that intersects their personal preferences.

Note: The EWall Application utilizes computation as a means to both expand upon the advantages and to resolve the limitations of Pena's traditional Problem Seeking methodology. The following chapter on EWall Cards introduces a computational version of Pena's physical card. The subsequent chapter on EWall Modules outlines a computational environment for the use and management of EWall Cards. The final chapter on EWall Settings illustrates a variety of possible combinations, processes, and settings for the EWall Modules.

EWALL CARDS

EWall Cards present a computational version of the physical cards proposed in Pena's Problem Seeking methodology (Illustration 1c). All components of the EWall Application use EWall Cards as the standard means for representing data and information. EWall Cards may also be evaluated as a potential replacement for file and desktop icons of computer systems. EWall Cards maintain many qualities of their physical counterparts while also introducing modifications and additions to the original card concept, layout and functionality.

Concept

As with physical cards, the making of EWall Cards continues to involve humans in the processes of converting, standardizing, abstracting, associating, and relating information. The computational nature of EWall Cards allows for additional functionality and alternative applications. Three main differences distinguish EWall Cards from their physical counterparts:

A first difference concerns the issue of card ownership. While Pena's Problem Seeking methodology stresses the advantages of cards as shared objects, the EWall project introduces functionality that encourages a stronger relationship between EWall Cards and their authors. For example, Pena's Problem Seeking sessions commonly engage specialists responsible for the creation of cards. The EWall Application provides the functionality for users to more easily engage in the process of creating EWall Cards. Users also control the access and distribution of their EWall Cards. Furthermore, the user histories of EWall Cards are carefully recorded to ensure authorship rights and to reconstruct the shared development of ideas. The ability to create, control and track EWall Cards is optional yet available for users that wish to obtain a sense of ownership over their contributions. Through the resolution of ownership issues EWall Cards can become the means for converting data, information and knowledge into a currency- or trading card-like format that can easily be collected, compared, organized, presented, stored, shared, exchanged, and sold. In this regard the EWall Card may be viewed as a "transitional object" [4] that a user can possess, view as something personal he knows and understands, and protect against modifications by the software and/or other users.

A second difference concerns the card contents. Card contents in Pena's Problem Seeking methodology usually represent hints and reminders of human knowledge such as ideas and concepts. EWall Cards share this particular quality but also allow for annotations, file attachments, and hyperlinks to data and information sources. With this additional functionality EWall Cards remain visually abstract while also providing fast access to relevant and more detailed information. This functionality allows for the concurrent management of pointers to human knowledge, computer files of various formats, as well as data and information located in remote locations.

A third difference concerns the card layout. Even though the traditional card layout did not suggest a specific location for graphics, keywords and textual information, cards created during programming sessions often complied with arbitrary standards to ensure the easy comparison of large numbers of cards. EWall Cards suggest a standardized layout yet do not exclude customized designs. The standardized layout of EWall Cards fosters the direct comparison of different types of information such as hyperlinks, emails, notes, and documents. Furthermore, the layout of EWall Cards encourages an objective evaluation and comparison of information based on content rather than type, source, author, and modification date.

Layout and Functionality

Both, the layout and the functionality of EWall Cards are highly modular. This means that almost every visual component and computational feature can be individually turned on and off, and that additional visual components and computational features can easily be integrated. Furthermore, the colors and fonts of all EWall Card components are customizable. The sizes of EWall Cards are also variable though working with only one size significantly simplifies comparisons and organization.

EWall Cards are visually subdivided into five segments that can expand and contract depending on their contents:

1. Icon Area

The Icon Area allows for the placement of graphical and textual material that can help users to quickly locate cards among many other cards as well as to memorize and recall the data, information or knowledge associated with cards. Users can copy and paste pictures and text from most computer applications into the Icon Area. Users can also drag and drop pictures and text from web browsers and file managers into the Icon Area. Furthermore, users can directly draw and write into the Icon Area. The background color of the Icon Area is also customizable allowing users to visually group and highlight cards.

2. Information Bar

The Information Bar accommodates the interface and visual indicators for complementary card functions. Every available card function is represented by a small rectangular box that contains an icon and in some cases a numerical counter. Three distinct background colors visually indicate the status of each function. The color gray is used for inactive functions, the color green is used for active functions, and the color red is used for functions that require attention. The indicators are user specific meaning that the icon colors of card copies may differ among users. Every function can be individually turned on and off. The Information Bar also allows for the customization and addition of functions specific to particular situations and work tasks. Many of the default functions are designed to support the use of cards in remote-collaborative settings (see Exchange Module).

The Comment Function allows for the addition of comments and annotations. Every new comment is automatically complemented with the authors' name, the date, and the time. The icon color indicates whether a card contains new or previously reviewed comments. The counter next to the icon displays the number of comments. This functionality is utilized as a means to explain and discuss card contents among collaborating users in distributed and asynchronous environments.

The Ownership Function allows users to replicate cards of other users. For example, if user A copies a card from user B then user A's card copy can not be modified and continues to adapt modifications from user B's original card. By taking ownership, user A creates and subsequently controls a new instance of the card that is independent of its original. The icon color indicates whether a card is original, a non-modifiable and adaptive copy, or a modifiable and non-adaptive copy. Adaptive card copies are primarily used to monitor changes and updates to cards while modifiable card copies allow for the reuse of cards in different contexts.

The Reference Function allows users to hyperlink or to attach related information. Users can complement cards with hyperlinks to web sites, files, directories, and executables. Alternatively, users can directly attach files and executables thus making cards independent of information stored in remote locations. The icon color indicates whether a card contains a new or previously reviewed hyperlink or attachment. This functionality enables users to directly compare and more conveniently administer information that resides in remote locations, receives frequent changes, and involves different file formats.

The Priority Function allows users to mark cards that they consider important. The icon color indicates whether a card is considered important or was considered important in the past. Card priorities are user specific meaning that this function cannot be used to signal the importance of cards to other users (see Votes Function).

The Vote Function allows collaborating users to exchange their opinions about the relevance of individual cards. Every user is given one supporting vote for each card. The icon color indicates the addition of recent and the presence of past votes. The numerical counter next to the icon displays the total number of supporting votes.

The Log Function provides users with a detailed record indicating the dates, times and names of all users that previously viewed, copied or modified the card. The icon color indicates recent and past log entries. The counter next to the icon displays the number of log entries. This functionality allows users to review the evolution, collaborative use, and authorship history of cards.

Illustration 2

Information Bar Functions

FUNCTION		Gray	Green	Red	left-click	right-click
✉	Comment	no comments	previously viewed comments	new comments	view and add comments	
♛	Ownership	original card	card copy dependent on original card	card copy independent on original card	take ownership	
◳	Reference	no reference	previously viewed reference	new reference	view reference	add reference
△	Priority	not important	important	important since a short time	mark important	
⊠	Vote	no votes	votes	recent votes	view votes	place vote
♘	Log	no log entries	log entries	recent log entries	view log entries	
⊸	Permission	distribution and access not restricted	distribution restricted	access restricted	restrict distribution and access	
☼	Notification	not a personal notification	personal notification confirmed	personal notification unconfirmed	send or confirm a personal notification	
⚑	Location	no location	location		view, select or change a location	

The Permission Function enables users to restrict the automatic distribution of cards and the access to card attachments. The icon color indicates whether restrictions are in place. Users of cards with locked attachments can request access from the card owners and the card owners can remotely unlock card attachments for specific users. This functionality supports the private use of cards, the sharing of cards with a selected group of people, and the trading of cards.

The Notification Function is used to announce cards to specific users. A card owner can add a list of addressees to a card. The icon color changes if the card is viewed by one of its addressees. Addressees confirm the receipt of a card by making a copy of the card or by reviewing its content. The card owner is notified once all addresses confirmed the receipt of the card. This functionality allows users to ensure that their cards are received and reviewed by the addresses. It is typically used for cards that contain instructions or important information.

The Location Function is used to associate cards with a geographic location on a map. The icon color indicates whether a geographic location has already been specified. This functionality is primarily used in conjunction with news messages that are commonly evaluated and categorized by source, content, creation date, and geographic relevance.

3. Date / Time Bar

The Date / Time Bar displays the date and time of the most recent card modification. The Date / Time indication is primarily used to reconstruct the chronology of contributions from different users and sources.

4. Author Bar

The Author Bar displays the name of the user who last modified the card or the information source the card has been copied from. The Author indication is primarily used to compare contributions from different users and sources.

5. Heading Bar

The Heading Bar allows users to complement cards with a brief description or some keywords. The ability to articulate the meaning of cards and card contents with only a few words is a unique human quality. The labeling of cards is essential for the management of large card assemblies and also allows the EWall Application to associate the individual words in the Heading Bar with other card variables. These associations can be used for a variety of tasks such as to present the user with a choice of illustrations for the Icon Area that best matches a particular set of words in the Heading Bar. The Heading Bar effectively complements the Icon Area by introducing language as an additional means for the abstract representation of data, information and knowledge. While the Icon Area is particularly useful for the visual navigation of large card arrangements, the Heading Bar is focused on supporting the quick and easy recollection of card meanings and contents.

Modules		Views			Interpretation Algorithms		Transformation Algorithms		Servers		
Workspace Module	WM	Workspace View	WV		Level I + II Algorithms	LI LII					
News Module	NM	News View	NV				News Algorithms	NA	News Server	NS	
Database Module	DM	Database View	DV		Level III + IV Algorithms	LIII LIV	Database Algorithms	DA	Database Server	DS	
Exchange Module	EM	Exchange View	EV				Exchange Algorithms	EA	Exchange Server	ES	
Visualization Module	VM						Visualization Algorithms	VA			

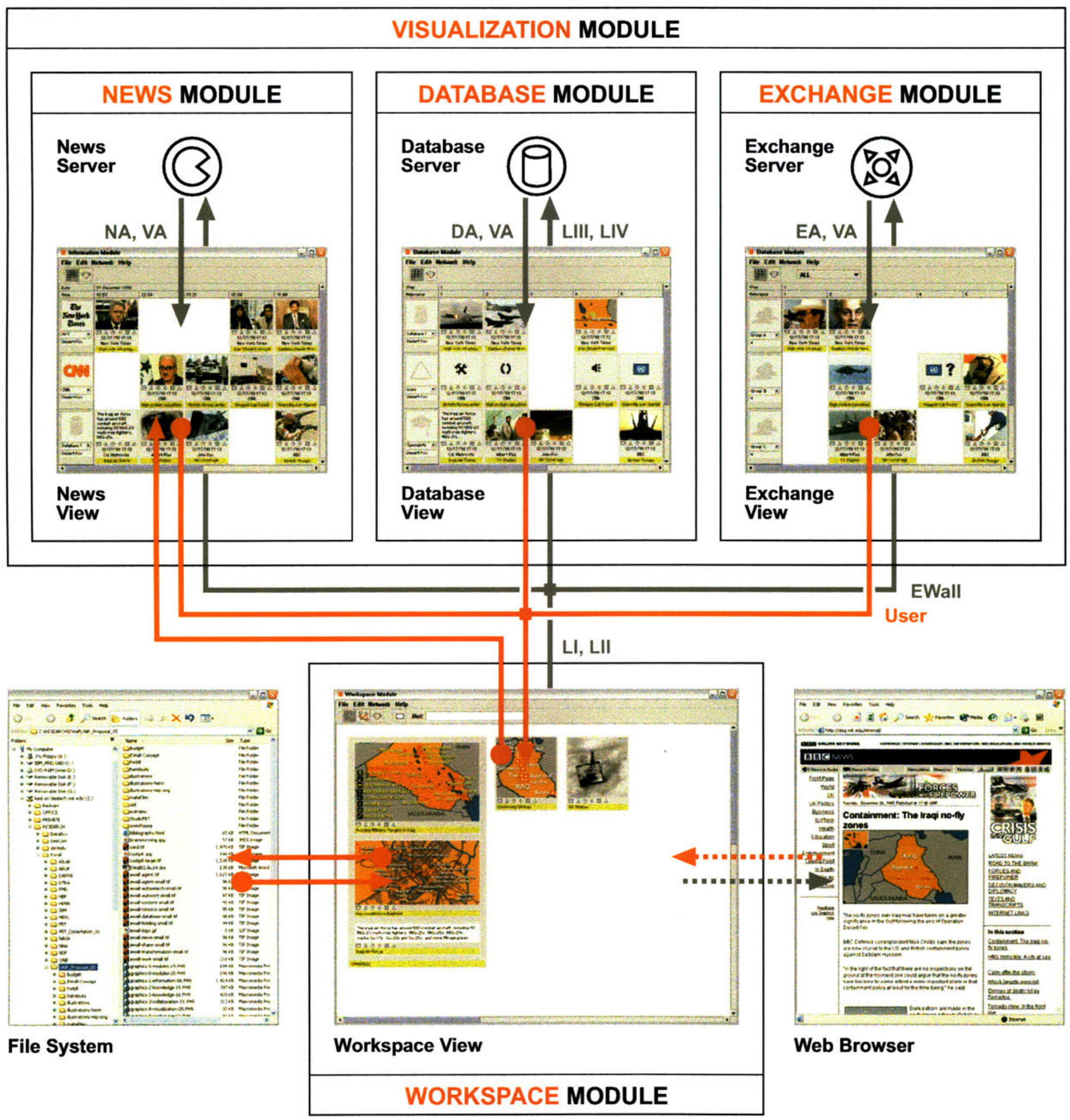


Illustration 3: Names, symbols, and abbreviations; Concept and module interoperability

EWALL MODULES

The EWall Application is divided into five modules. The modules present users with the environment and functionality for the use and management of EWall Cards. The modules are aimed at making administering, monitoring, collecting, exchanging, and visualizing information more intuitive. The modules support the manual, semi-automatic, and automatic creation of EWall Cards as well as the search, exchange, and organization of EWall Cards. The separation of the EWall Application into multiple modules also assures the easy customization, maintenance and scalability of the EWall Applications as well as its connectivity with other software applications.

Concept

The EWall Modules dynamically adapt to users and changing circumstances while also trying to stimulate human sense-making activities. The goal is to make humans more situation-aware by exploiting the human capacity for visual problem solving and to make the EWall Modules more situation-aware by inferring meaning from the observation of human activities. In other words, the EWall Modules promote an environment in which computers and people make each other more knowledgeable. This is implemented by virtue of advocating a circular information flow between the users and the EWall Modules. Through this circular information flow both the users and the EWall Modules gradually develop a shared understanding of particular tasks and continuously adapt to changing processes and circumstances.

Illustration 3 and 4 explain the concept of the circular information flow between the users and the EWall Modules: 1. The Workspace Module provides the user with an initial Workspace View, which is an empty computer window for the creation and organization of EWall Cards. 2. The Workspace Module formulates probable relations among cards based on the analysis of the user's spatial card arrangement. 3. Both the cards and the relations are sent for further analysis to the server components of the News, Database, and Exchange Module. 4. These three modules provide the user with one computer window each (News, Database, and Exchange View) that display information relevant to the card arrangement in the user's Workspace View. The News View presents recent information whose relevance has not yet been determined, the Database View presents mature information whose relevance can be estimated, and the Exchange View presents information produced by collaborating users. This particular separation of relevant information corresponds with three essential sense-making activities: to keep up with task related news, to gather task related information from available sources, and to collect task related opinions and views from trusted individuals. The News Module may be compared with a newsstand for the acquisition of recent information, the Database Module with a library for the exploration of older information, and the Exchange Module with a meeting space for the discussion of task specific information. The Visualization Module allows the user to choose from a variety of ways in which cards are arranged and visualized in the News, Database, and Exchange Views. 5. The user can copy cards from the News, Database, and Exchange Views into the Workspace View. 6. Every card copy indicates a successful suggestion and is reported back to its respective module thus fostering additional suggestions based on similar considerations. This feedback mechanism allows the individual modules to improve the accuracy of their suggestions by incrementally adjusting to particular users, tasks, and circumstances. While users review information suggested by the EWall Modules, the EWall Modules advance through user observation and feedback. Thus, both the users and the EWall Modules maintain some influence over each others actions.

Illustration 4

Circular Information Flow

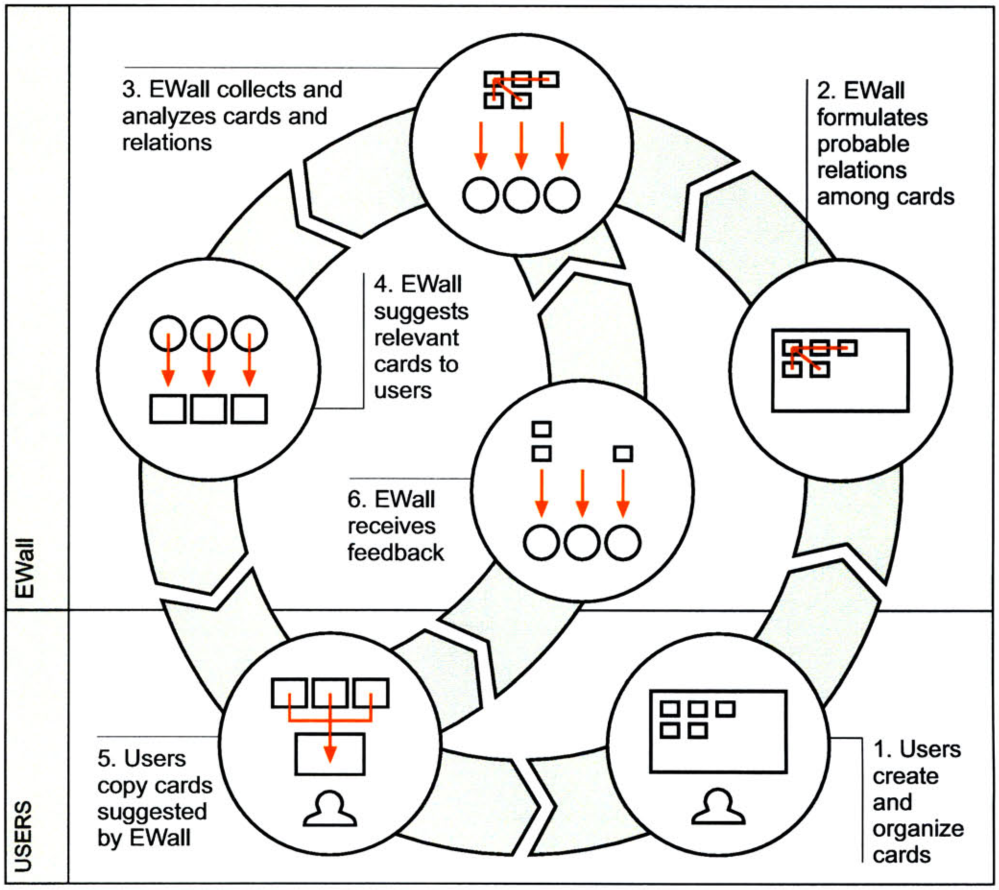
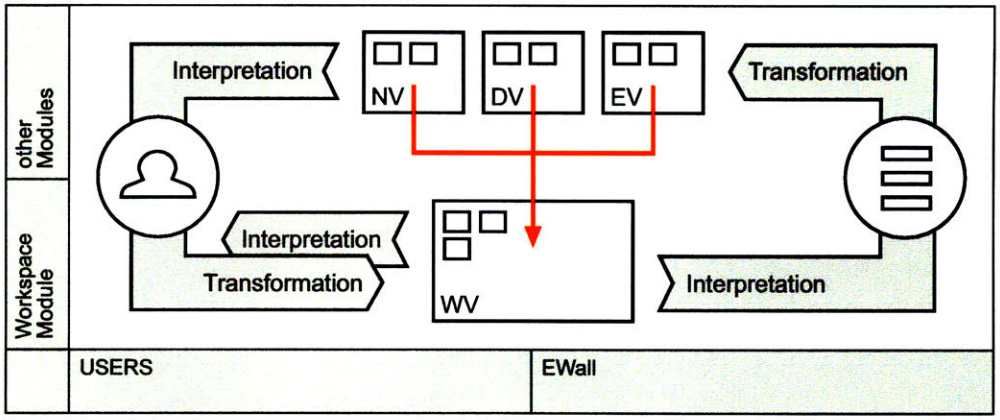


Illustration 5 compares the interdependent work tasks of users and EWall Modules during sense-making activities. Users focus primarily on collecting, creating, and organizing cards. The users' understanding of the relations among cards advances through the continuous interpretation and transformation of card arrangements. In this context the word Interpretation refers to the (mental) activity of analyzing and comprehending the card contents and the relations among cards. The word Transformation refers to the (physical) activity of manipulating card arrangements to explore and detect alternative relations among cards. While the Workspace Module provides users with the means to effectively manage and comprehend information, the remaining four modules support the accumulation of information. The EWall Modules interpret, combine and analyze the card arrangements of multiple users. The subsequent transformation of the combined card arrangements produces selections of potentially relevant cards for every individual user. These selections of cards are presented in separate Views to clearly distinguish the windows controlled by the users and the EWall Modules. By default, relevant cards are arranged from left to right in the order of relevance. A user with no time or interest might entirely disregard the suggestions, a user with very little time or interest might only consider suggestions that are displayed on the very left, and an explorative user might review a large number of suggestions by evaluating the string of suggestions towards the right. The suggestions are meant as an inspiration and subject to the users' interpretations.

Illustration 5

Interpretation and Transformation



Technical Research Areas

The EWall Modules encompass an integrated set of tools and methods for sense-making activities that correlate with a variety of technical research areas in Information Visualization, Information Communication, and Information Management (Illustr. 6):

1. Information Visualization

Sense-making activities deal with the formulation and evaluation of complex relations among large amounts of data, information and knowledge. As there is a limit to a human's mental ability to memorize and relate newly acquired information, sense-makers often rely on external representations such as notes, diagrams and sketches. These representations are commonly organized, arranged and compared (bricolage of information) as a basis for comprehension and inspirational thought. The EWall Modules provide users with the means to abstract, standardize, and spatially arrange information. The approach rests on the premise that these activities engage untapped aspects of human cognition and knowledge retrieval thus allowing for the resolution of more complex sense-making tasks. The EWall Modules also promote an unconstrained and flexible computational environment for the management of information and the simultaneous use of different information formats. Furthermore, the EWall Modules encourage the creative interpretation of information by engaging users in the process of visualizing information in various different ways.

2. Information Communication

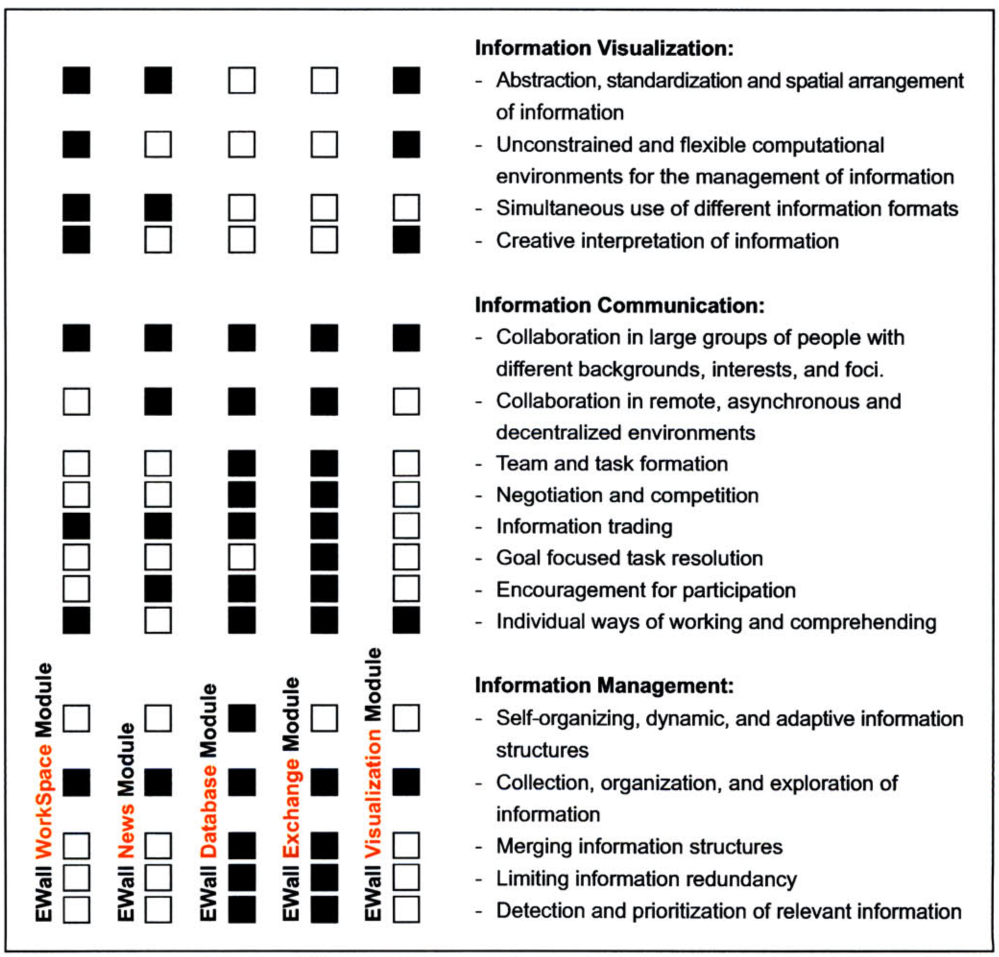
Complex sense-making tasks commonly require the collaboration among people with different areas of expertise, individual ways of working, and diverse agendas. While collaborative sense-making activities often lead to more innovative insights and conclusions they can also significantly slow down the sense-making process. In other words, a large number of collaborators might contribute more knowledge to a sense-making task yet also spend more time on exploring intersecting views and negotiating conflicting opinions. The EWall Modules support collaborative sense-making activities in large groups of people with different backgrounds, interests, and foci in remote, asynchronous, and decentralized environments. The EWall Modules introduce mechanisms for team and task formation, negotiation and competition, information trading, and goal focused task resolution. The EWall Modules also promote a computational environment that encourages participation and that sustains individual ways of working and comprehending.

3. Information Management

Sense-making activities often accumulate large amounts of information. The time it takes to categorize information increases with the growth of information. The best possible categorization for a particular set of information usually dynamically changes with every addition or modification and also depends on the preference of individual users. While the maintenance of dynamic and user specific categorizations can become very time-intensive, the use of static and user independent categorizations is often inflexible and confusing. The EWall Modules operate with self-organizing, dynamic, and adaptive information structures that allow users to easily collect, organize and explore information. Multiple information structures can be merged and data redundancy limited through the reuse and recombination of information. The EWall Modules also introduce a variety of algorithms for the detection and prioritization of relevant information.

Illustration 6

Relevance for Information Visualization, Communication, and Management



Note: The following five chapters introduce the functionality of the individual EWall Modules. The underlying computational mechanisms (Algorithms) are explained in Section B. The table in Illustration 3 correlates the EWall Modules and Algorithms.

1. Workspace Module

The Workspace Module provides users with an environment to create and spatially arrange EWall Cards. The Workspace Module also introduces technology for the detection of both explicit and implicit relations among spatially arranged cards. The Workspace Module may be used by itself yet is required for the operation of the remaining four modules.

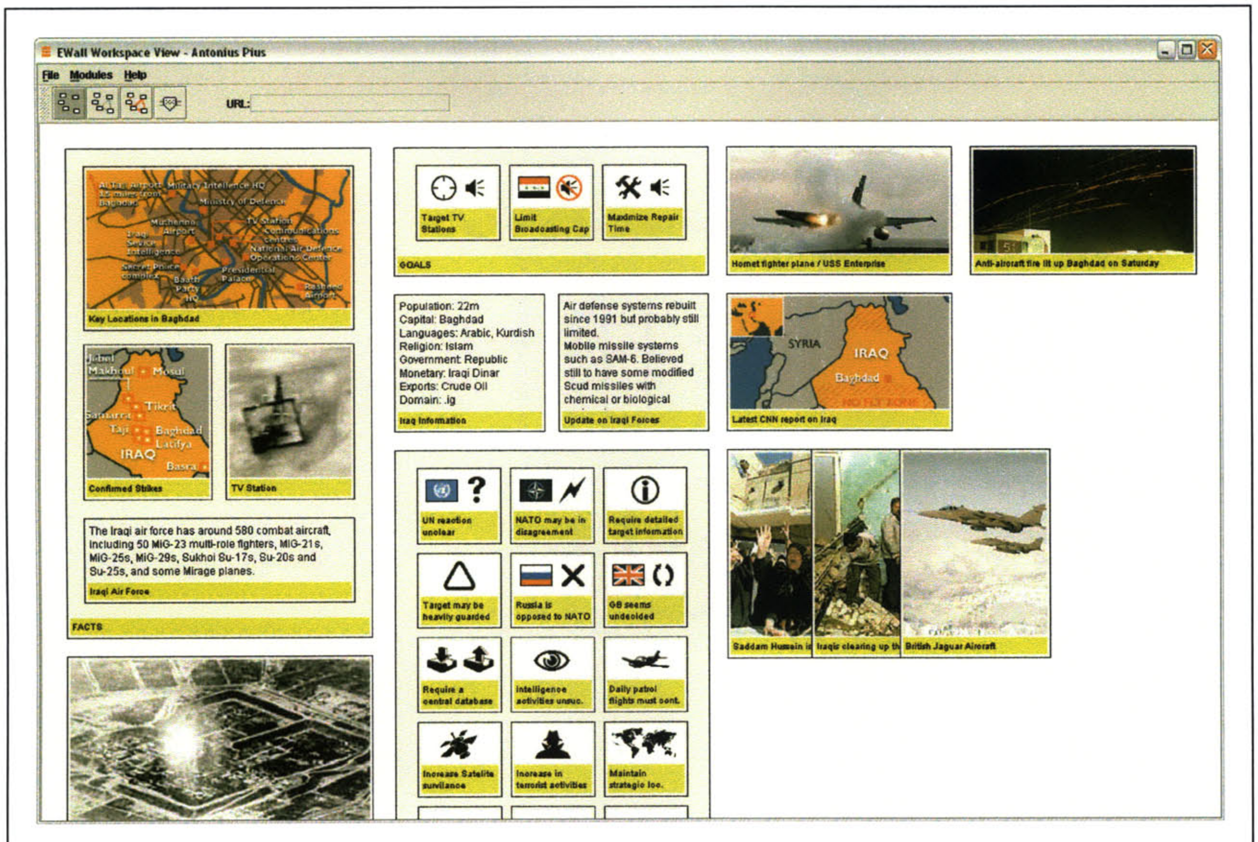
Concept:

The Workspace Module presents a computational alternative to Pena's physical card arrangement. While physical card arrangements require users to draw and pinup cards, computational card arrangements allow users to create and spatially arrange cards on a computer canvas. The Workspace Module is also open to a wide variety of applications and collaborative settings superceding those suggested in Pena's Problem Seeking methodology. For example, the Workspace Module could be used individually or collaboratively as a tool for preparing projects, reports, papers and presentations, for maintaining to-do lists, or for organizing documents such as picture, audio and video files. Furthermore, the Workspace Module could be utilized as a digital board for announcements, news, alerts, questions, and discussions.

The handling of physical and computational card arrangements differs in a number of ways:

A first difference concerns the collaborative use of card arrangements. The shared development of physical card arrangements effectively supports collaborative sense-making activities by enhancing situational awareness and group identity. A physical card arrangement evolves through the discussion and negotiation of contributions from multiple individuals and consequently emerges to a shared product indicating areas of common understanding and intersecting views. The emerging content and context of physical card arrangements can be compared with organizations and cities whose configurations materialize over time through the collaboration of individuals contributing to a common goal while also protecting their personal interests. Computational card arrangements share the collaborative advantages of their physical counterparts yet allow for a greater variety of collaborative settings by introducing the functionality for the use of shared and interconnected card arrangements by both locally present and remotely distributed individuals.

A second difference concerns the management of card arrangements. The static nature of physical card arrangements makes it difficult for users to easily rearrange and copy cards, to store and transport card arrangements, and to study the development of card arrangements. Computational card arrangements are not affected by these limitations. For example, computational card arrangements can be



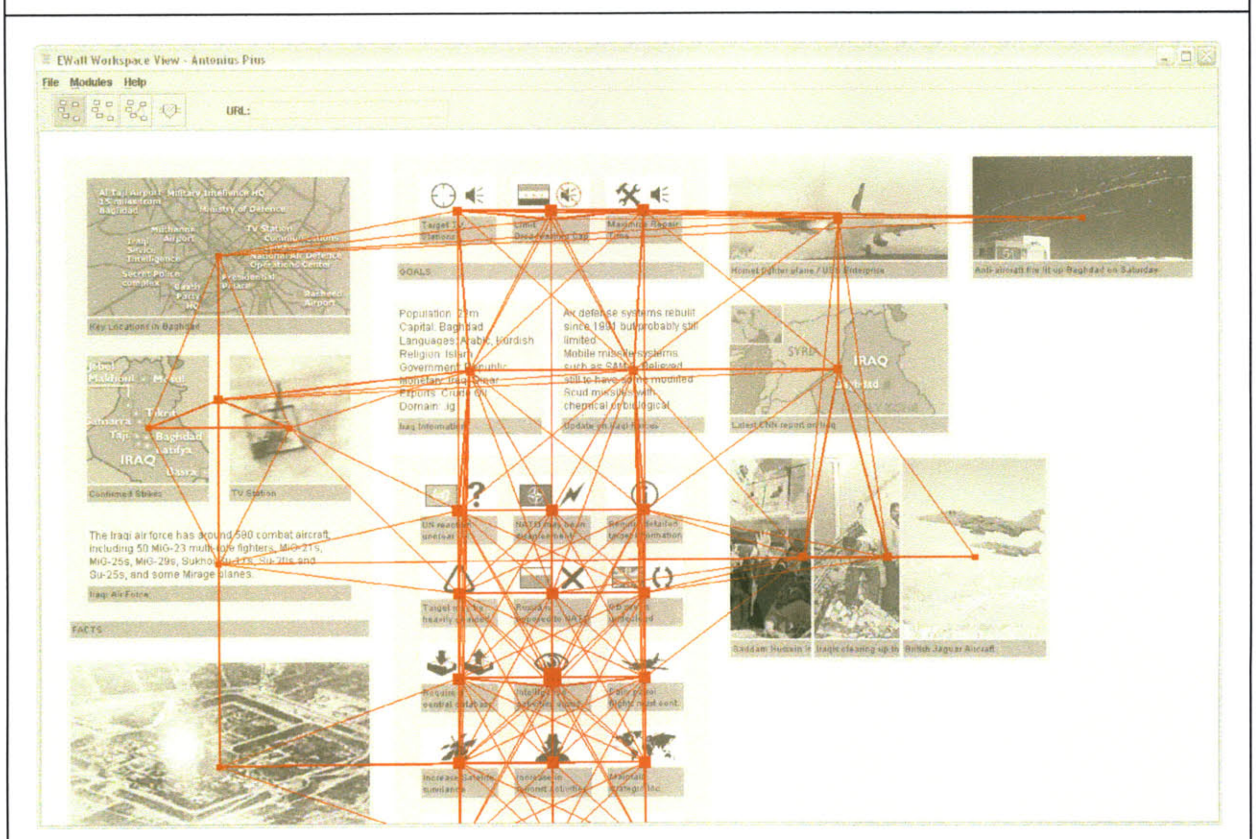


Illustration 7a: Workspace View; 7b: Visualization of explicit and implicit relations among cards

administered in databases and communicated over the Internet. Computational card arrangements can also be saved and previous arrangements can be restored, allowing users to freely rearrange cards without the potential loss of previous compositions. Furthermore, cards and card contents can easily be copied and their authorship history traced back in time.

A third difference concerns the visualization of card arrangements. In a physical space, the number of cards that can be simultaneously displayed is limited by the dimensions of the available pinup space. The use of multiple pinup boards or entire walls allows for the display of many hundreds of cards. In a digital space, the number of computational cards that can be displayed simultaneously is limited by the computer's processor speed and available memory as well as the number and the resolution of the computer displays or projectors. While a computer with a speed of 3 GHz and 1024 MB of memory can handle several thousand cards, a computer display or projector with a resolution of 1600x1200 pixels can only display up to 100 cards in full quality. In other words, while computational card arrangements can be very large only a small number of cards can be viewed simultaneously. Consequently, navigating large computational card arrangements requires users to select, scale, or rearrange particular sections of card arrangements thus compromising among the organization, range, number, and quality of visually accessible cards as well as the scope of cognitively comprehensible relations among cards.

A fourth difference concerns the ways in which people interface with card arrangements. Groups of individuals can interface with physical card arrangements by using their hands to jointly arrange, add, and remove cards. In contrast, computational card arrangements are indirectly controlled by a single user who operates the software application. Research in interactive display technologies such as DynaWall [5], the Designers' Outpost [6], the Interactive Workspaces Project [7], aire [8], or Augmented Surfaces [9] suggest a verity of solutions for the collaborative use of large scale displays with specific focus on technologies for gesture and object recognition. These solutions will eventually allow users to directly interface with computational card arrangements.

Interface and Functionality:

The user interface of the Workspace Module is referred to as the Workspace View (Illustration 7a). The Workspace View presents users with an empty canvas for the creation and arrangement of EWall Cards. The Workspace View introduces several options for the explicit grouping of cards: A first option is to increase the size of one card so as to accommodate several other cards inside its boundaries. Moving a card will drag along all cards within its boundaries. A second option is to overlap cards. Moving a card inside a cluster of overlapping cards will drag along the entire cluster of cards. A third option is to create links (rubber lines) between cards. Links may be labeled and contain arrowheads on one or both ends. The Workspace View also offers a snap mechanism for the easy alignment of cards as well as an XML export option for the storage of card arrangements and arrangement templates. Users can drag and drop cards from the Views of other modules into the Workspace View. Users can also drag and drop cards from the Workspace View to the computer desktop or file

Illustration 8: Screen shots of an animated Workspace Module demonstration

system (and vice versa) thus converting EWall Cards into regular computer files. This functionality has a variety of applications such as to exchange EWall Cards by email or to convert EWall Cards for the use with other software applications. Finally, users can maintain more than one Workspace View to create a space for cards whose relevance has not yet been determined or to separate personal from publicly accessible cards.

The Workspace Module introduces technology for the detection of explicit and implicit relations among spatially arranged cards (see Illustration 7b). Explicit relations among cards are assumed if users overlay or link cards. Implicit relations are determined based on the relative location of cards using concepts such as proximity and alignment. Implicit relations are also determined through the comparison of card attributes such as author names and modification dates. The computational mechanisms for the detection of explicit relations are referred to as Level I Algorithms and the computational mechanisms for the detection of implicit relations are referred to as Level II Algorithms (see Section B). The Algorithms are modeled based on observations about how humans establish and identify relations among spatially arranged objects. The Workspace Module uses these Algorithms to extract contextual information from the users' spatial card arrangements. The availability of both the contents (cards) and the contexts (relations among cards) of the users' spatial card arrangements is essential for the remaining four modules to effectively search, prioritize and arrange cards for specific users as well as to combine the card arrangements of multiple users.

The interface and functionality of the Workspace Module show similarities with a variety of software applications such as Inspiration [10], SmartDraw [11], Visio [12], Tinderbox [13], OmniGraffle [14], QuestMap [15] and Notification Collage [16]. The main differences between these applications and the Workspace Module is the layout and functionality of the EWall Cards, the focus on grouping rather than connecting cards, the detection of relations among spatially arranged cards, as well as the complementary functionality provided by the remaining four modules.

Illustration 8 shows screen shots of an animated Workspace Module demonstration. Users can create EWall Cards in Workspace Views (1). The Icon Area and the Heading Bar of EWall Cards can contain text (2). The Icon Area of EWall Cards can also contain pictures. Texts and pictures from other applications may be copied directly into EWall Cards (3, 4). Users can hyperlink EWall Cards with web sites (5). A red icon indicates that the card is hyperlinked (5). A mouse click on the icon opens the associated web site in a web browser (6). A green icon indicates that the card owner already visited the associated web site (7). Drawing applications may be used to complement EWall Cards with hand sketches (8). Users can resize (9), move (9), customize (10, 11), and erase (12) EWall Cards. Users can also copy EWall Cards from their News, Database, and Exchange Views (13). The users' Workspace Views are analyzed after every modification (14). The analyses are sent to all connected News, Database, and Exchange Servers. The Servers update the respective Views of individual users with information relevant to what they currently are working on (15).

2. News Module

The News Module keeps users informed about recent additions and modifications to information sources such as web sites, shared computer directories and databases. The News Module consists of a News Server that monitors, collects and abstracts information as well as a News View that displays the information with EWall Cards. Users can copy EWall Cards from their News Views to their Workspace Views. Users can also send EWall Cards from their Workspace Views to the News Server. (see Illustration 3)

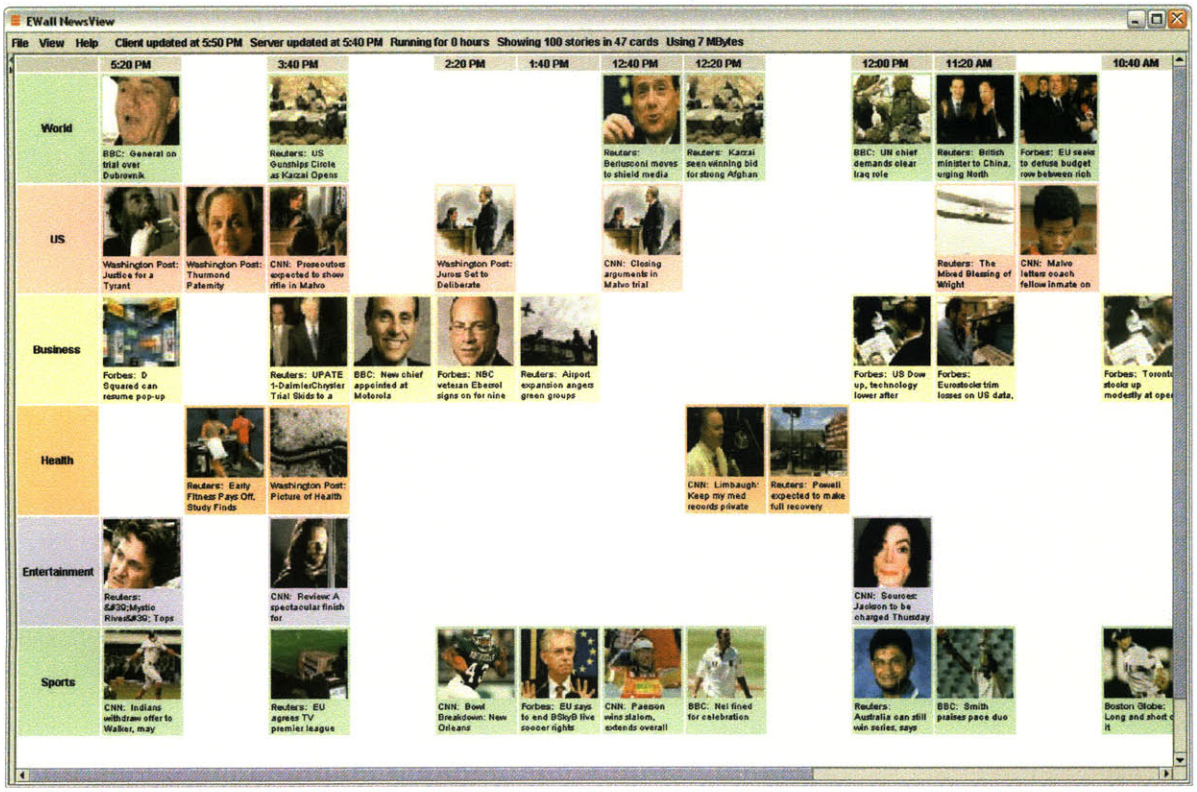
Concept:

The purpose of the News Module is to help users stay current with news that potentially affects their work tasks. The continuous monitoring of work task related news allows humans to quickly react and adapt to changing circumstances. For example, a stockbroker might decide to sell stocks because he just received a notification indicating a downturn in consumer confidence. The problem is that large amounts of news form large numbers of news sources are difficult to keep track of. The various steps involved in detecting work task related news include 1. the selection of appropriate news sources, 2. the continuous monitoring of selected news sources for additions and modifications, 3. the filtering of additions and modifications for relevant news, 4. the organizing of relevant news, and 5. the evaluation of relevant news. The News Module supports this process by monitoring user selected information sources, by filtering additions and modifications for user specified keywords, and by organizing the results based on age and information source.

The News Module is focused on alerting users of the availability of new information rather than on determining the value and relevance of new information. Estimations about the value and relevance of new information are often vague and consequently best left to the individual users. However, in some cases assumptions about the value and relevance of new information can be derived from the information content, the type and credibility of the information source, the expertise and reputation of the author, or the creation date of the information. In contrast, information that has existed for some time offers additional hints about its value and relevance. These hints emerge over time through the organization and collaborative use of information (see Database, Exchange, and Visualization Module).

Interface and Functionality:

The News Module consists of a News Server and a News View (see Illustration 3). Users can maintain their own News Server or connect to the News Servers of other users. The News Server monitors and collects additions and modifications from user specified information sources. Examples of information sources include web sites, shared computer directories, databases, and EWall Servers. Examples of content retrieved from information sources include news, emails, chat messages, phone messages, comments, requests, questions, answers, notifications, announcements, alerts, sensory inputs, and EWall Cards. The News View collects information from multiple News Servers, filters the information for user specified keywords, and



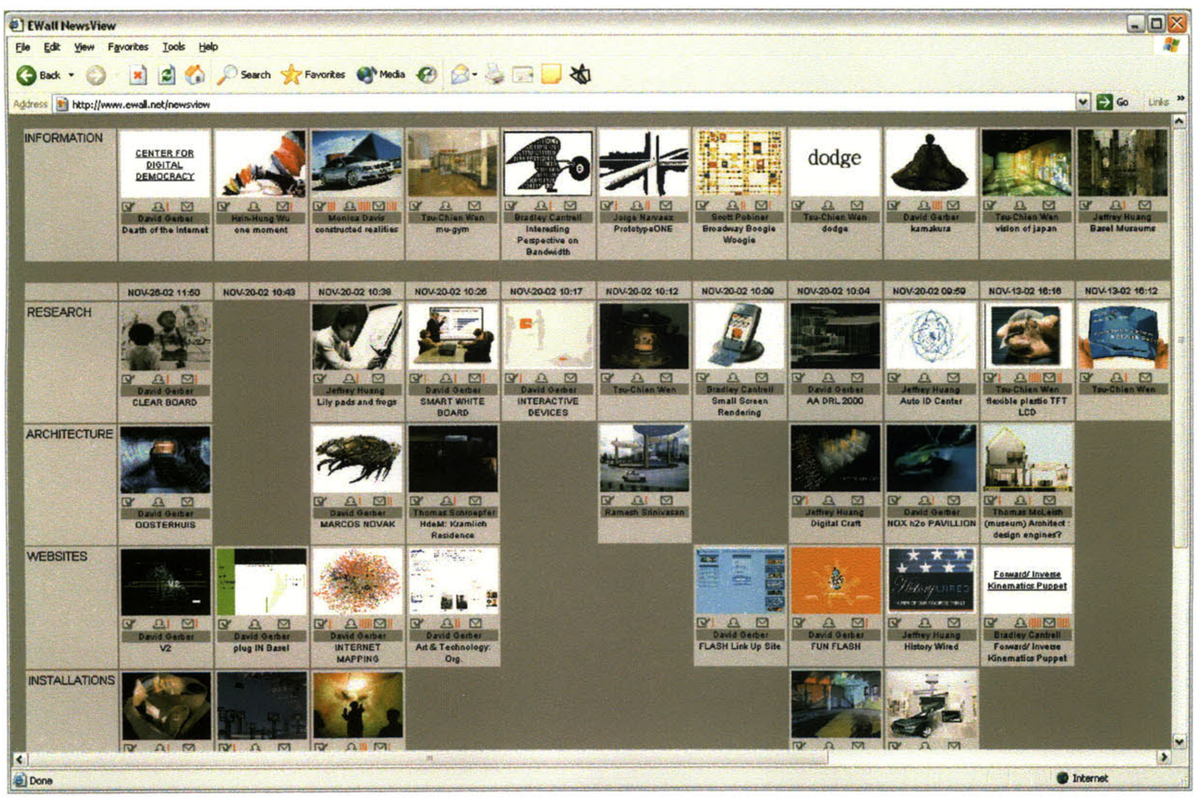


Illustration 9a: News View; 9b: News View (web based version)

organizes the information by age and information source. Users can view the results in a variety of different ways (see Visualization Module). By default the results are displayed with EWall Cards, arranged in a table, and organized by time in horizontal and by information sources in vertical direction. New cards are inserted on the left pushing existing cards to the right. The time scale is irregular and optimized to display the largest number of cards possible. The rows automatically expand and contract depending on the number of cards. Illustration 9a shows an instance of News View that displays additions to ten different web based newspapers. The rows separate news categories such as World, Health, Politics, and Business. Illustration 9b shows an instance of a web based version of News View [17] that displays the contributions of multiple collaborating users conducting a field research. The News View is best leveraged in combination with the Workspace View. EWall Cards can be copied from the News View to the Workspace View thus allowing users to easily collect, compare and organize new information.

The News Module determines the value and relevance of new information for specific users by comparing the streams of new information with the contents on the individual users' Workspace Views. Because the evaluation of new information is often inaccurate the results are only visualized in a modest way. EWall Cards with a red Priority Icon (see EWall Cards) hint new information of possible value and relevance to the user. The computational mechanisms for determining the value and relevance of new information are referred to as News Algorithms (see section B).

3. Database Module

The Database Module introduces a database whose structure evolves and dynamically adapts via the collaborative effort of individual users creating, collecting, organizing, exchanging, and exploring EWall Cards. The Database Module consists of a Database Server that autonomously copies, analyzes and organizes EWall Cards from the Workspace Views of multiple users as well as a Database View that provides individual users with EWall Cards relevant to what they are currently working on. Users can copy EWall Cards from their Database Views to their Workspace Views (see Illustration 3 and 10a).

Concept:

The Database Module autonomously copies EWall Cards from the Workspace Views of multiple users. The Database Module may be used as an automatic backup application capable of restoring the Workspace Views of any user at any moment in time. However, the Database Module is not conceived as a simple recording device but as a self-structuring and dynamic database that not only collects but also analyzes and organizes EWall Cards. Thus, the Database Module may be compared with an independent observer capable of recognizing, memorizing, and correlating the contributions of many individuals. The database consists of cards and relations among cards. Every card is related to at least one other card and multiple relations can exist between pairs of cards. The database content increases through the addition of new cards and the database context changes through the addition of new relations.

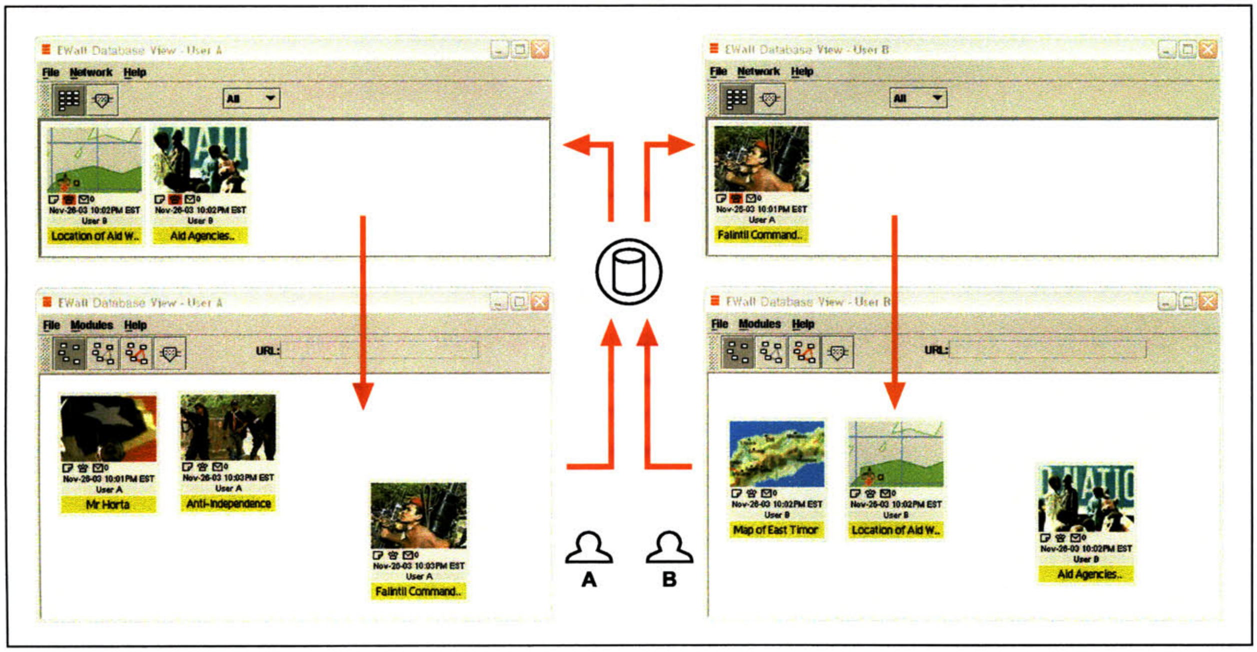


Illustration 10a: Database View and Workspace View; 10b: Screen shots of an animated Database Module demonstration

The Database Module not only assimilates cards and relations from the Workspace Modules of individual users but also constructs its own relations based on database activity and structure. The continuous addition of cards and relations allows the database structure to evolve and dynamically adjust to changing users and work tasks. The Database Module utilizes the database structure by searching and prioritizing the database contents for information relevant to specific users.

The Database Module introduces a database structure that suggests similarities with semantic networks. The difference is that the Database Module does not supplement relations with semantic information, combines relations between node pairs, and analyzes the database structure independent of the node contents. The Database Module's autonomous search for user relevant information may be compared to a conventional search engine that continuously executes keyword searches for recently typed or spoken words (e.g. IBM's MeetingMiner [18]). The difference is that the Database Module's search is initiated by a particular user's card arrangement and that the results are prioritized based on the constellation of relations in the database.

The Database Module's unconventional database structure offers several benefits:

A first benefit concerns the self-structuring ability of the database. No prior knowledge about the future use of the database is required as its structure emerges over time through the continuous addition of relations. Furthermore, the self-structuring ability of the database permits for the incremental and dynamic combination and organization of large numbers of cards created by different authors, at different times, for different purposes, and under different circumstances. This functionality is especially beneficial for organizational knowledge management as it allows for the administration of information generated by collaborating and non-collaborating members of one or more organization as well as the subsequent analysis of the database structure for overlapping interests and the availability of specific expertise.

A second benefit concerns the continuous and dynamic adaptation of the database structure to changing users and work tasks. The database structure is defined through the constellation of relations and changes with the addition of new relations. Imagine for example a database that contains three cards A, B and C as well as one relation between card A and B and one relation between card B and C. The later addition of a second relation between card A and B will make the relation between card A and B appear "stronger" than the relation between card B and C thus causing a change to the database structure. As relations are primarily created based on user activities, the database structure is likely to change and gain strength in areas that contain information relevant to current users and work tasks. Dynamic adaptation also effectively deals with data expiration. Old and less-commonly accessed relations no longer gain strength and eventually loose their competitive advantage over newer relations. The disadvantage with the dynamic adaptation is that the database looses its adaptability with increasing number of relations (similar to the ways humans loose their adaptability to new concepts with age). This is because relations that remain popular over a long period of time can gain enough "strength" to supersede over more recent and potentially more important relations.

A third benefit concerns the ways in which the database is searched and explored. The database contents are not only searched through the conventional use of query and sorting mechanisms but also through the exploration of directly and indirectly related cards. The relations among cards help users to detect meaningful and inspiring correlations among cards as well as to develop a contextual understanding of the database contents. For example, cards related to many other cards are more likely to contain information of general importance and common interest. Also, card pairs connected with many relations often indicate a strong relevance, communality or dependency. Users under time pressure might limit their explorations based on obvious factors such as directly related cards and cards that are connected with many relations. Explorative users might also consider indirectly related cards and cards that are connected with only few relations. As some relations are established based on database activity (see Section B, Level III Algorithms, Footprint and Query), a search or exploration itself may cause a structural change. Thus, the repetition of a search or exploration may potentially produce different results.

A fourth benefit concerns the reuse and recombination of information. As some relations are established based on the reuse and recombination of card contents (see Section B, Level II Algorithms, Replication), a relation between two cards might indicate that some of the content on the first card was copied from the second card or that the first card is a modified version of the second card. While these particular relations converge with other types of relations, they do influence the database structure in ways that help users to track the development and authorship history of cards, examine the contexts in which cards were created, and recognize cards that contain redundant and complementary information.

A fifth benefit concerns the preservation of the database history. The database preserves its history in a unified rather than in a divergent manner. A divergent preservation refers to conventional methods of preserving a database history by creating occasional backups. The advantage of a divergent preservation is that database stages are accurately preserved and that users can compare different database stages as well as undo changes. The disadvantage of a divergent preservation is that backups accumulate quickly and that the analysis, comparison and exploration of large numbers of backups are often time-intensive and confusing. A unified preservation means that the database history is exclusively derived from the most recent database state through the analysis of card attributes (e.g. modification dates, authors, etc.) and accumulating relations. To allow for an inclusive reconstruction of the database history, database contents cannot be erased but only marked inactive. The database history is examined by visualizing different aspects of the database contents. For example, a particular visualization could highlight weakly related database segments which are often interpreted as assemblies of cards that have received little or no attention. Another example is to fade cards whose attributes have not changed for a long period of time thus allowing users to easily recognize and focus on database modifications. While the historical analysis of a database with unified preservation may be less accurate, it is often sufficient to approximate some of the more essential aspects.

Interface and Functionality

The Database Module consists of a Database Server and a Database View (see Illustration 3). Users can maintain their own Database Server or connect to the Database Servers of other users. The Database Server autonomously copies and combines the contributions of all participating users. The Database Sever may contain contents from many independent and collaborating users, and from many related and unrelated projects. The Database View provides individual users with database contents relevant to what they are currently working on. A user's Database View is updated after every modification to his Workspace View. Users can view the results in a variety of different ways (see Visualization Module). By default the results are displayed with EWall Cards and arranged from left to right in the order of relevance. The horizontal scroll bar allows users to explore less relevant EWall Cards towards the right. The exploration of less relevant EWall Cards is important as it is often inspiring and may trigger the recollection of relevant knowledge.

The Database Server includes computational mechanisms for complementing the database structure with additional relations. The relations are conceived based on the analysis of database activity and structure. Database activity is registered if the database receives new additions and if the database is searched or explored. The database structure refers to the constellation of relations at a particular moment in time. The computational mechanisms for establishing relations based on database activity are referred to as Level III Algorithms and the computational mechanisms for establishing relations based on the database structure are referred to as Level IV Algorithms (see Section B). The Database View includes computational mechanisms for selecting and prioritizing EWall Cards relevant to specific users. The selection and prioritization process is based on the analysis and comparison of the users' Workspace Views as well as the contents and structure of the database. The computational mechanisms for selecting and prioritizing EWall Cards are referred to as Database Algorithms (see section B). The computational mechanisms of both the Database Server and the Database View can be optimized through the manual addition, removal and adjustment of algorithms. The Database Server and the Database View also contain functionality for the self-optimization of their algorithms. The self-optimization is based on indirect user feedback and obtained every time a user copies a card from the Database View to the Workspace View. The feedback value reflects the estimated user relevance of the card. For example, a card copied from the very left of the Database View resonates in a high feedback value while a card copied from the near-right resonates in a low feedback value. The feedback value determines whether to reinforce or weaken the algorithms that dominated the selection and prioritization of the card.

Illustration 10b shows screen shots of an animated Database Module demonstration. The demonstration introduces two users whose Workspace and Database Views are connected to the same database (1). Whenever a user creates a new card on his Workspace View the Database Module copies the card to the database (2, 3). Relations established by the Workspace Module are also copied to the database (4, 5, 6). The Database Module also complements the database with relations (7). The continuous addition of cards and relations expands the network of cards (8). The Database View retrieves, prioritizes and displays cards relevant to each user (9).

4. Exchange Module

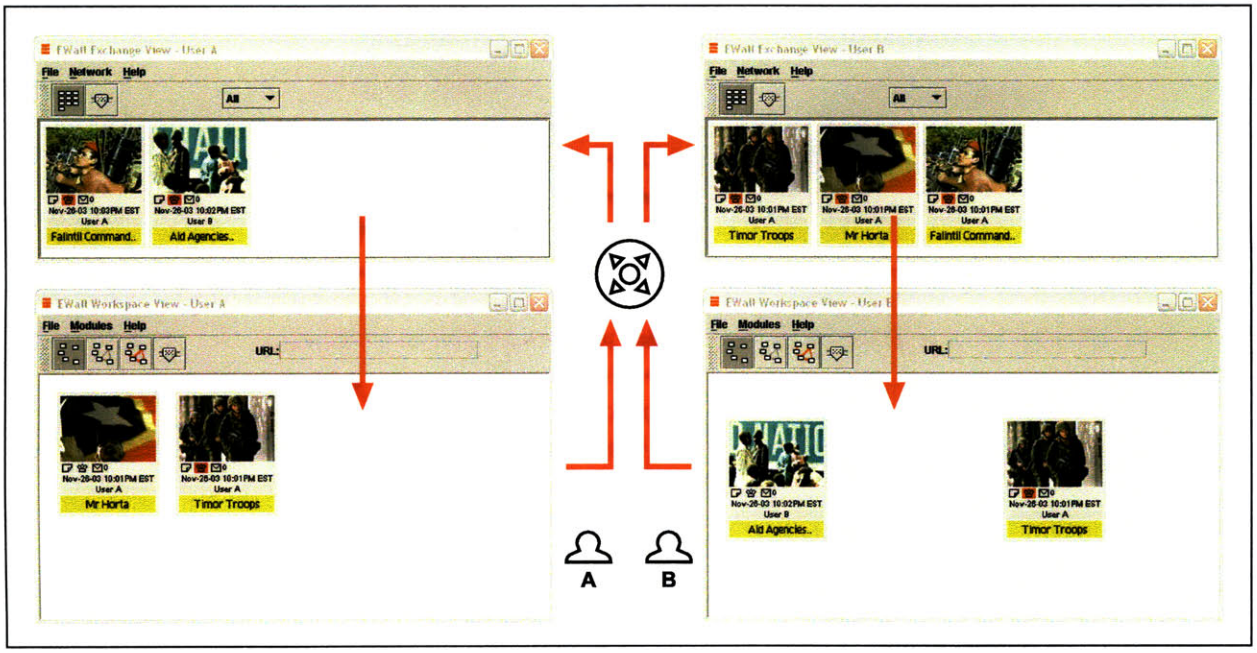
The Exchange Module allows users to copy EWall Cards from the Workspace Views of other users and enables the collaborative functionality built into EWall Cards (see EWall Cards). The Exchange Module consists of an Exchange Server that interconnects the Workspace Views of multiple users and an Exchange View that displays the card arrangement from the Workspace View of one participant or the combined contents from the Workspace Views of all participants. The combined contents are organized in the order of relevance for every individual user. Users can copy EWall Cards from their Exchange Views to their Workspace Views (see Illustration 3 and 11a).

Concept:

The Exchange Module supports the remote, asynchronous, and anonymous collaboration of users with varying levels of involvement and different foci. More specifically, the Exchange Module supports the administration and distribution of large amounts of shared information as well as the indirect collaboration among large numbers of locally present and remotely distributed participants. Unlike the News and Database Module, the Exchange Module does not focus on the accumulation of information but the exchange of information among collaborating users. The Exchange Module is intended to complement rather than replace existing tools for collaboration.

1. Administration and distribution of large amounts of shared information:

Collaborating users deal with both unique and shared information [19]. Unique information refers to knowledge and information held solely by individual users. Shared information is accessible to all collaborating users at all times. Research on the collaborative impact of unique and shared information indicates that the use of shared information greatly benefits the qualitative outcome of collaborative tasks. For example, Stasser and Titus [20] tested the collaborative impact of unique and shared information by asking a small group of participants to rank hypothetical job applicants. During a first test series every participant was provided with all available information about every applicant. During a second test series the information was distributed among the participants. The participants correctly determined the best applicant in 83% of all cases during the first test series and only in 24% of all cases during the second test series. A variety of theories correspond with the outcome of these two test series. For example, Gigone and Hastie [21] believe that most consideration is given to shared information and that shared information is most influential towards the final group judgment. Wittenbaum et al. [22] noted that the presence of shared information allows participants to mutually confirm each other's expertise. Furthermore, Michael Schrage's [23] statement that "people respond to what's just been said, not something said earlier" supports the notion that the promotion of unique information by individual participants is only of temporary consideration to other participants, and consequently does not have as much of an impact as shared information that is permanently available to all participants. Finally, Lavery et al. [24] observed that unique information is not primarily used "to exchange information but to aggregate member preferences into a consensual group judgment" thus neglecting



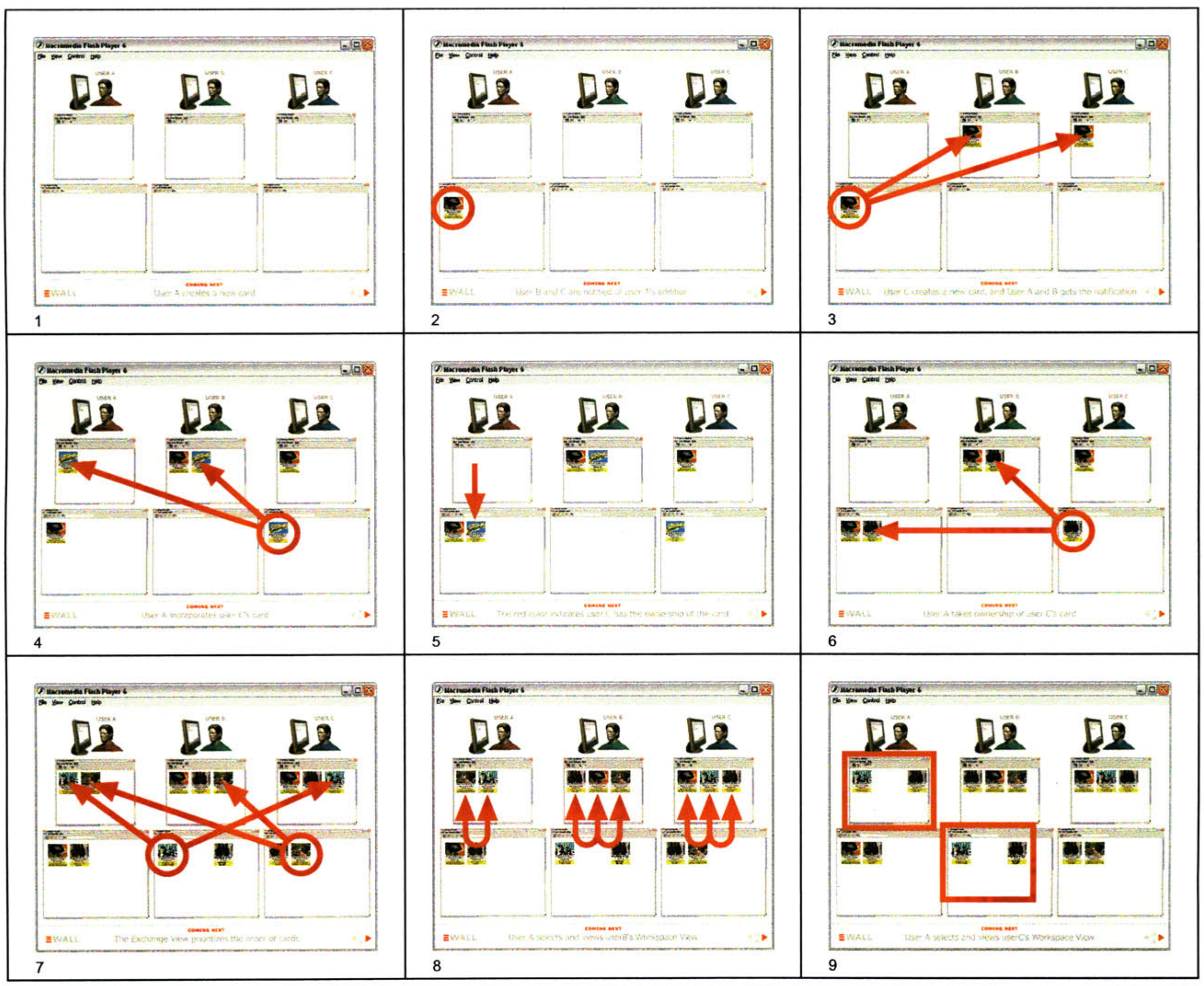


Illustration 11a: Exchange View and Workspace View; 11b: Screen shots of an animated Exchange Module demonstration

opportunities for leveraging unique information for the greater benefit of establishing a shared understanding.

The limited effect of unique information is especially evident in remote and asynchronous collaboration. Remote and asynchronous collaboration is less focused on direct user interaction and consequently offers fewer opportunities for the propagation of unique information. A possible solution is to increase the amount of shared information for remote and asynchronous collaboration. However, the conversion of unique into shared information as well as the administration of large amounts of shared information is often very time intensive. More specifically, the conversion of unique into shared information requires information contributors to evaluate uniquely held information for items that potentially benefit the community. The evaluation of shared information requires information beneficiaries to search shared information for relevant items. Furthermore, if the information contributors and the information beneficiaries are not in continuous communication (common in remote and asynchronous collaboration) then the selection of unique information is based on the subjective evaluation of the information contributors and consequently may not adequately reflect the particular needs of the information beneficiaries.

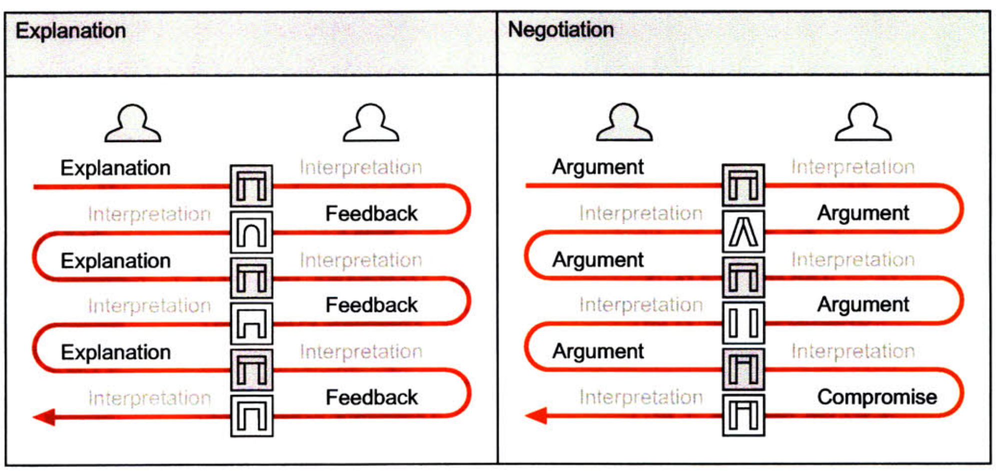
The Exchange Module supports the administration and distribution of large amounts of shared information. By default, information created on a user's Workspace View is considered unique. By enabling users to interconnect and access each other's Workspace Views, this information becomes shared. (Users can restrict the access of individual EWall Cards or maintain a second Workspace View for private use.) Furthermore, the Exchange Module averts information overload by sorting shared information in the order of relevance for every individual user. Consequently, the Exchange Module does not require information contributors to evaluate their Workspace Views for information that should be shared with other users and provides information beneficiaries with the means to quickly explore large amounts of shared information for relevant items.

2. Indirect collaboration among large numbers of participants:

Collaboration may happen directly or indirectly. Direct Collaboration refers to situations in which participants remain present at all times, communicate synchronously, and focus on one issue at the time. Direct collaboration is more common in physical settings although video conferencing systems, application sharing software, and instant messaging tools offer alternatives. The advantages of direct collaboration are that participants can easily exchange their views and learn about their unique perspectives. The disadvantages of direct collaboration include the small number of participants and the unused potential of individual participants during discussions that do not correspond with their backgrounds, expertise, interests or foci. Indirect Collaboration refers to situations in which individual or small groups of participants work independently on tasks that comply with a common objective. Indirect collaboration is common for tasks that span over long periods of time and that involve large numbers of participants. Indirect collaboration is more common in virtual settings but is also present in physical settings. Physical settings that best support indirect collaboration include open work environments such as unpartitioned office spaces, design studios, and control centers. Such environments allow for sporadic, incidental, and dynamically changing interactions among individuals. For example, an increase in chatter in a trading room often indicates the occurrence of an event of common importance. The main advantage of indirect collaboration is that it allows for the remote, asynchronous, and intermittent participation of people with varying levels of involvement. Indirect collaboration also provides opportunities for anonymous contributions that are less influenced by social factors such as reputation, prestige and organizational status. A typical disadvantage of indirect collaboration is that communication mostly occurs among pairs of individuals. This means that information is usually not broadcasted but passed from participant to participant. Thus, information often remains distributed and only slowly propagates across the network of participants. Another disadvantage of indirect collaboration is that individuals commonly know little about the people they communicate with. This often leads to the misinterpretation of information. Michael Reddy illustrates the misinterpretation of information with his Conduit Metaphor [25], an environment in which several individuals live by themselves in different and unique worlds. The individuals only communicate with each other through the exchange of text messages. Due to the uniqueness of the individual worlds, the text messages make little sense to their recipients. For example, instructions on how to use an axe have little applicability in a world with no wood. Reddy's metaphor suggests that the interpretation of information greatly depends on the contexts that information is created in as well as the backgrounds of the information contributors and beneficiaries. Thus, the challenge in indirect collaboration is to establish a shared understanding between information contributors and beneficiaries to reduce misinterpretations and to ensure an effective information exchange.

Illustration 12

Explaining and Negotiating



The means of establishing a shared understanding between information contributors and beneficiaries differs in direct and indirect collaboration. Direct collaboration is usually sender controlled meaning that the information contributors select and customize information for specific information beneficiaries. For example, sales people promote their merchandise in ways that appeals to particular customers. Direct collaboration allows information contributors and beneficiaries to explain and negotiate their views, opinions and suggestions. Thus, a shared understanding often evolves within the context of a conversation (see Illustration 12). Indirect collaboration is usually receiver controlled meaning that the information beneficiaries are expected to retrieve and comprehend information. For example, if a person working in a trading room becomes aware of an increase in chatter (see earlier example) then the person might approach some of the people involved in the chatter to find out about what happened and inquire as whether this particular event is of relevance. In indirect collaboration, a shared understanding materializes through the information beneficiaries' increasing knowledge about the information contexts and contributors. For example, a market economist might evaluate a stock market forecast by investigating the recent economic development and the credibility of the information source.

Direct and indirect collaboration are not exclusive and predetermined styles of collaboration but ideally emerge and coexist during collaborative activities. As direct collaboration usually only allows for one person to talk and one issue to be considered at a time, individual participants often have to wait for an opportunity to introduce their ideas and suggestions or wait until a discussion shifts to an issue that matches their interests and expertise. Thus, a directly collaborating group of people may decide to temporarily separate into multiple indirectly collaborating sub-groups. Such sub-groups may consist of people of common or complementary backgrounds, expertise, interests and foci. The sub-groups may independently consider a common task or work in parallel on multiple sub-tasks. The challenge of switching from direct to indirect collaboration is to create and associate sub-groups and sub-tasks. Often the optimal configuration of sub-groups and sub-tasks is dynamically changing and may require participants to occasionally switch sub-groups or to simultaneously contribute to more than one sub-task. The challenge of switching from indirect to direct collaboration is to merge the contributions of individual participants and to help the newly formed group to quickly develop a shared understanding, a discussion focus, and a common goal.

The Exchange Module supports indirect collaboration among a large number of participants in various ways: First, the Exchange Module connects the Workspace Views of all collaborating users. This functionality enables individual users to review and copy contents from the Workspace Views of their colleagues. It allows users to work independently while simultaneously contributing to, and benefiting from, the work of their colleagues. The functionality also allows users to share a Workspace View and copy contents between shared and personal Workspace Views. This allows for the separation of information that reflects the shared understanding of all users from information that benefits the comprehension of individual users. Secondly, the Exchange Module combines the contents on all users' Workspace Views and displays the results in the order of relevance customized for every individual user. This functionality enhances user and information awareness by allowing individual users to monitor and evaluate the contributions of a large number of participants and by providing instant access to the Workspace Views associated with particular contributions. The exchange and organization of user contributions also reduces the necessary amount of verbal communication and introduces alternative means for the promotion and negotiation of individual contributions. Thirdly, the Exchange Module compares the Workspace Views and activities of all users to determine overlaps in interests and foci. This analysis not only provides a basis for the dynamic formation of sub-groups and sub-tasks but also for progressing towards a coherent understanding and consensus. The analysis also supports the recognition, comparison, and association of differing views and opinions among users.

Interface and Functionality

The Exchange Module consists of an Exchange Server and an Exchange View (see Illustration 3). Users can maintain their own Exchange Server or connect to the Exchange Servers of other users. The Exchange Server more effectively supports the collaboration among large numbers of participants yet also benefits small groups of people. The Exchange View displays the card arrangement on the Workspace View of one specified user or the combined contents on the Workspace Views of all collaborating users. Users can view the combined contents in a variety of different ways (see Visualization Module). By default the results are displayed with EWall Cards and arranged from left to right in the order of relevance. The horizontal scroll bar allows users to explore less relevant EWall Cards towards the right. A user's Exchange View is updated after every modification to his Workspace View. The Exchange Module also enables the collaborative functions on EWall Cards (see EWall Cards). The Comments, Votes, Access Logs, and Personal Note functions help users to promote, discuss, and negotiate the contents associated with EWall Cards. The Ownership and Access Rights functions help users to control the distribution of EWall Cards.

The computational mechanisms for arranging EWall Cards in the order of relevance are referred to as Exchange Algorithms (see section B). The order of relevance is customized for every individual user and determined based on the analysis and comparison of the users' Workspace Views as well as the collaborative use of EWall Cards. Like the Database View, the Exchange View allows for the manual and self-optimization of its Algorithms (see Database Module).

Illustration 11b shows screen shots of an animated Exchange Module demonstration. The demonstration introduces three users whose Workspace and Exchange Views are connected to the same Exchange Server (1). The cards on the Workspace View of a user are displayed on the Exchange Views of both collaborating users (2, 3, 4). Users can copy cards from their Exchange Views to their Workspace Views (5). Unless a user takes ownership of a card copy, the card copy cannot be modified and continues to adopt modifications of the original card (6). The continuous addition of cards on the Workspace Views increases the number of cards displayed on the Exchange Views (7). The cards on the Exchange Views are arranged in the order of relevance for every individual user (8). Users can view the card arrangements of other users on their Exchange Views (9).

5. Visualization Module

The Visualization Module visualizes the contents presented by the News, Database and Exchange View in a variety of different ways (see Illustration 3 and 13). Users can easily combine and quickly switch between visualizations.

Concept:

Information sharing is common if not essential for all forms of collaboration. Shared information is usually stored, organized, and presented in ways that allows individual participants to easily find the pieces of information they are looking for. For example, physical information could be spatially arranged on a pin board or organized in a filing cabinet. Similarly, virtual information could be organized in a database or on a web server. Any organization of information has its virtues and deficiencies. The best possible organization of information differs and dynamically changes depending on the type of information, the user, and the circumstances. Thus, a variety of factors are to be considered when organizing information:

1. The information access time depends on how well the organization of information corresponds with the backgrounds and foci of current users and circumstances. An organization of information is never perfect meaning it cannot account for all possible users and circumstances. This is why people often disagree over how information should be organized and experience difficulties retrieving information organized by others. A common solution is to customize the organization of information for specific users and specific circumstances. For example, architects, engineers, and builders use different plans of a building specific to their area of expertise. An architect's plan might highlight room sizes and furniture, an engineering plan might highlight the material details of walls and ceilings, and a builders plan the locations of pipes and electrical installations. Likewise, different street maps for a city may feature different information such as tourist attractions, bus routes, and traffic directions. Furthermore, retail stores may sell the same merchandise yet attract different customers due to their particular presentations of products.

2. A customized organization of information can divert from interesting accidental findings such as the detection of related and alternative information. For example, the manual search for a book may traverse through several shelves and in some cases lead to the discovery of books that may prove more relevant or more interesting. A computational search for a particular book in a library on the other hand is typically unambiguous and points to an exact location. Consequently, a relaxed organization of information may allow for a more creative exploration of information.

3. An organization of information can hint possible relations among the individual pieces of information. For example, books located in close proximity on shelves are more likely to contain similar or related information. However, one particular organization of information seldom unveils all possible relations. For example, books may be organized either by author, subject, language, age, value, or reader. Thus, multiple different organizations of an information space may unveil additional relations and consequently allow for a more comprehensive analysis. For example, architects

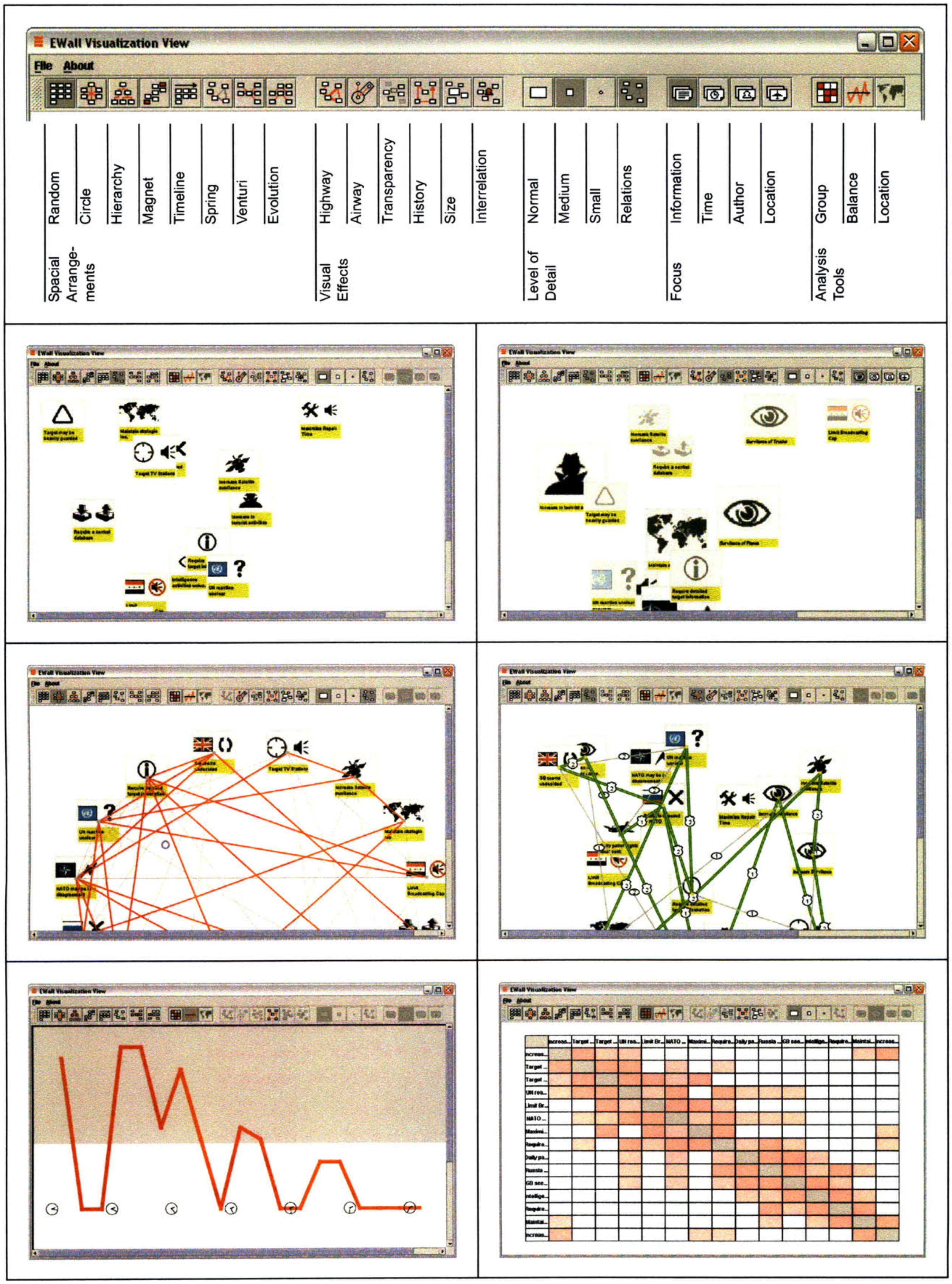


Illustration 13: Visualization Module Interface

commonly use multiple representations simultaneously to better understand the properties and spaces of a building design. Such representations might include models, renderings, animations, plans, sections, and elevations. Because every individual representation only visualizes a few aspects, architects combine these representations into a coherent mental model to gain a more complete understanding of the building design.

4. The use and organization of information can also allow for statistical interpretations. For example, the number of books in a particular section of a library may indicate the popularity or complexity of a specific subject. Furthermore, books in bad shape may have been reviewed by a large number of people and consequently are more likely to contain information of common interest. In some cases, the analysis of an information space is more relevant than the information itself. For example, a network analyst may be interested in server access frequencies rather than server contents. Similarly, a store manager may be interested in what products sell best rather then what these products are used for.

The Visualization Module acts as a filter between a human and an information space. Different lenses and combinations of lenses are used to change the visual representation of an information space, to emphasize particular aspects of an information space, or to visualize less obvious aspects of an information space. The use of these lenses offers several cognitive benefits: 1. Users can choose lenses that present information in familiar ways or that best support their understanding of an information space. 2. Users can quickly alternate among different lenses to provoke creative and inspiring interpretations of an information space. 3. Users can compare the visualizations produced by different lenses to develop a more complete understanding of an information space. 4. Users can investigate the distribution, development and collaborative use of information with lenses that are specifically designed for the analysis of information spaces.

Interface and Functionality

The Visualization Module complements the interface of the News, Database, and Exchange View with functionality to customize the visual presentation of EWall Cards. The Visualization Module may also be used as a standalone application for the visual presentation of non-EWall data. The different functions are divided into five groups and represented by one button each (see Illustration 14): Spatial Arrangements organize EWall Cards in different ways. For example, EWall Cards may be organized chronologically or hierarchically. Only one Spatial Arrangement can be selected at a time. Visual Effects change the appearance of EWall Cards and relations. For example, the modification dates of EWall Cards may be represented with different shades. Any combination of Visual Effects can be used with any Special Arrangement. Analysis Tools provide statistical information about a particular selection of EWall Cards and relations. Level of Detail settings are used to abstract EWall Cards and hide relations. Focus settings are used to specify groups of properties by which EWall Cards are organized and analyzed. Examples of properties include the number of relations associated with particular EWall Cards and the locations that EWall Cards were created at or are referring to. Any combination of Focus Settings can be used with any Spatial Arrangement or Analysis Tool.

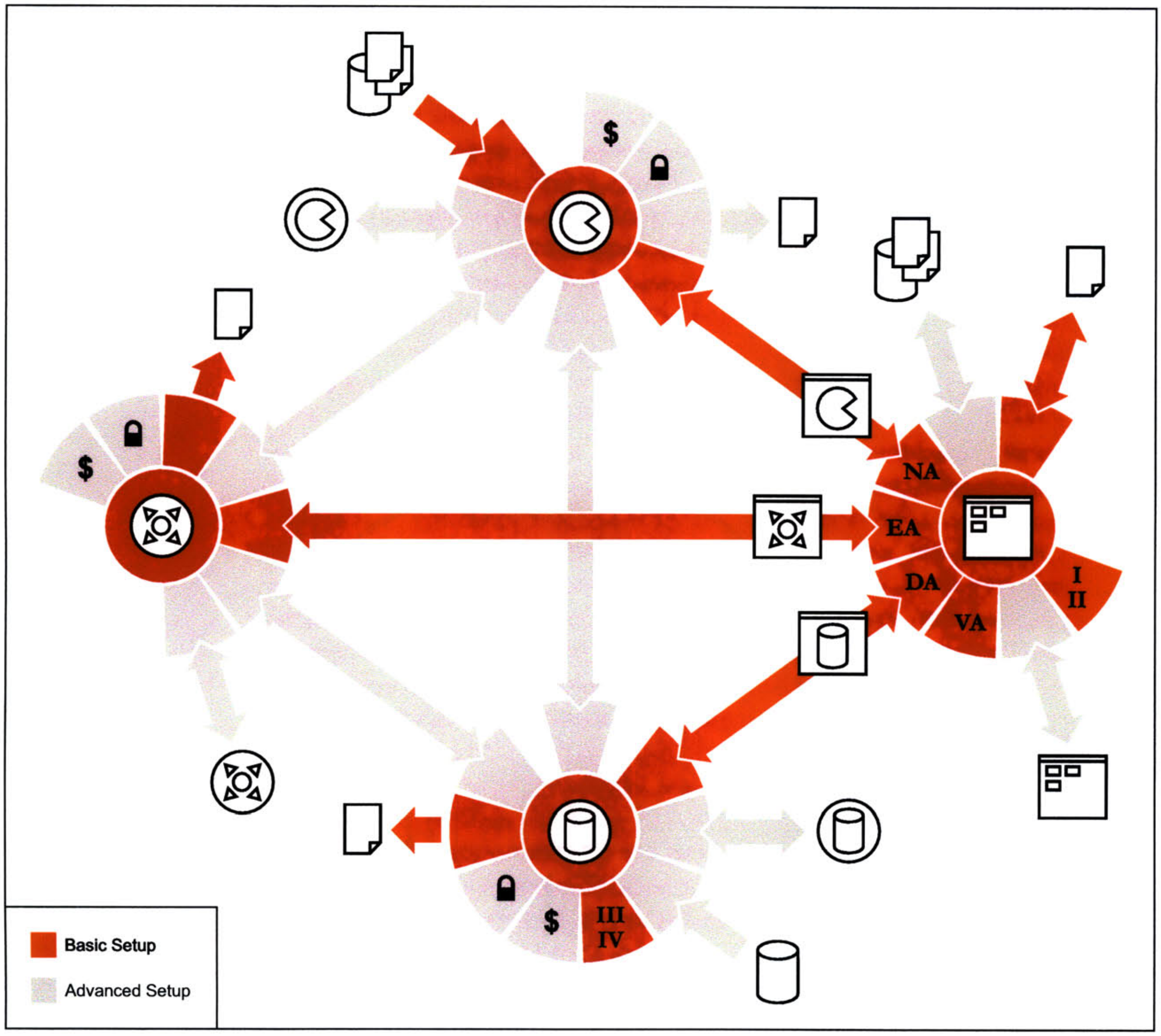
The computational mechanisms for the Spatial Arrangements, Visual Effects, and Analysis Tools are referred to as Visualization Algorithms (see section B). The Visualization Module includes functionality for the parsing, compatibility testing, and scaling of non-EWall data. The parsing functionality identifies data attributes such as hyperlinks, dates, authors, and headings. The parsing functionality also identifies data structures such as groups and hierarchies. The compatibility testing functionality analyzes data files for their compatibility with particular visualizations. The scaling functionality separates the data into smaller portions suitable for particular visualizations and window sizes.

EWALL SETTINGS

The modular components of the EWall application can be combined for various collaborative settings, users, work tasks, and work processes. The EWall application is scalable and adjustable through the dynamic addition, removal, and recombination of components and connections among components.

Illustration 14a shows possible combinations of EWall components (see Illustration 3 for names, symbols and abbreviations). The Workspace View can be used by itself or in combination with a News View, a Database View, an Exchange View, or additional Workspace Views. The News, Database and Exchange Views require a connection with at least one their Servers. The selection and visualization of EWall Cards on the News, Database and Exchange Views is customizable through the use and modification of News, Database, Exchange, Visualization, Level I and Level II Algorithms. The News, Database and Exchange Servers can be interconnected. Interconnected Servers of the same type share information. For example, a News Server featuring BBC News interconnected with a News Server featuring CNN News allows for the retrieval of BBC and CNN news from either Server. Interconnected Servers of different types exchange information. For example, a News Server could monitor additions and modifications to a Database Server or information passing through an Exchange Server. Similarly, a Database Server could automatically archive all additions to a News Sever and information passing through an Exchange Server. All three Servers contain functionality for the import and export of data as well as for the management of transactions ($ Symbol) and access rights (Lock Symbol). The Database Server additionally allows for the use and modification of Level III and IV Algorithms.

Users can act as information providers and information receivers, operate multiple server and client components, and connect their server and client components with the server components of other users. The network of information providers and information receivers represents a trading place for information. This trading place may be seen as a collective memory or intelligence that emerges through the distribution of information and through the decentralized network of autonomously acting individuals. The trading place allows for a wide variety of possible user roles and for the dynamic formation and coexistence of different types of communities. Examples of user roles include team leaders and participants, employers and employees as well as teachers and students. Examples of communities include project-



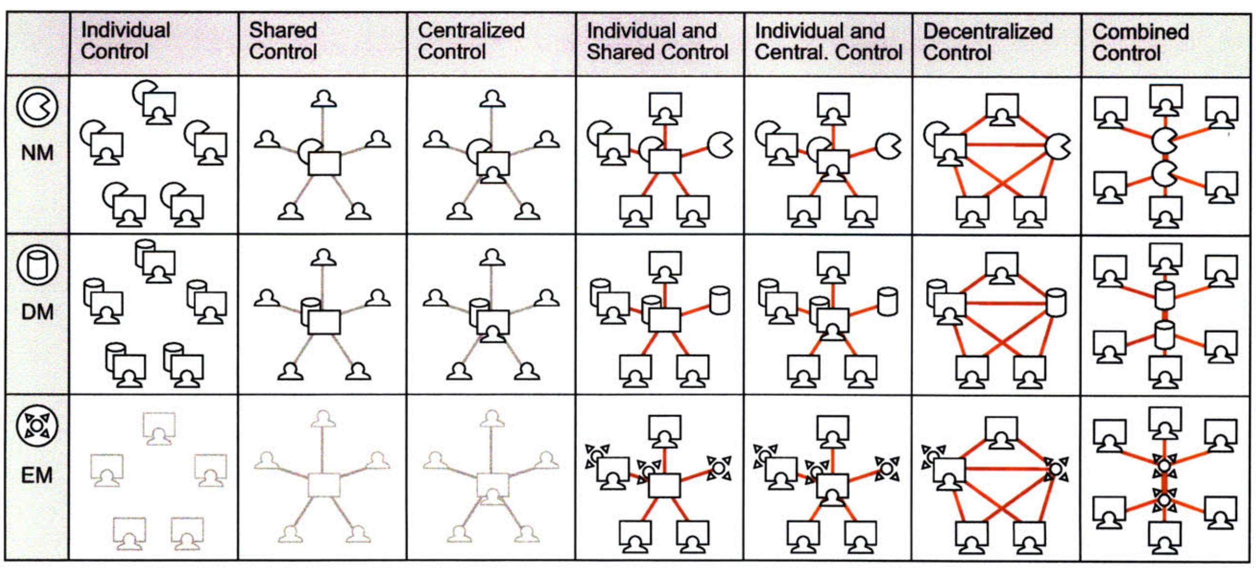


Illustration 14a: Possible connections among the modular components of the EWall application; 14b: EWall Settings

based communities that share a common objective, practice-based communities that share similar specializations and expertise, and social-based communities that share similar interests or environments [26]. Every Server or Workspace View can potentially define a community. Furthermore, the flexible interconnectivity of Servers and Workspace Views allows for the easy creation, combination, and separation of communities. Communities may be physical or virtual, and public or private. For example, the use of a Workspace View on a shared display suggests a physical community and the use of an Exchange Server for the collaborative resolution of a work task implies a private community. Communities may also be distinguished by means of describing their organizational structures or the authorities and responsibilities of individuals. Combinations of EWall components for specific organizational structures are referred to as EWall Settings and are differentiated by the varying degrees of user control (see Illustration 14b):

1. Individual Control

Individual Control applies to situations in which people neither communicate nor collaborate but use EWall components for the resolution of personal work tasks. For example, a researcher might be using a Workspace View to organize the contents of a paper he is working on. The researcher might also utilize a Database Server to automatically archive contents on his Workspace View as well as a Database View to explore related contents from his earlier papers.

2. Shared Control

Shared Control applies to situations in which participants share authority meaning that the communication among collaborators is not orchestrated by one particular individual. Shared Control is typical for peer meetings in which participants assemble around a table to exchange their views and expertise on a particular subject. Shared Control assumes that users collaboratively operate EWall components. For example, the participants of a meeting might project a Workspace View onto a large screen to jointly articulate and communicate ideas.

3. Centralized Control

Centralized Control applies to situations in which one person or a small group of people is in command and guides or assists collaborating participants. Centralized Control is typical for educational training, presentations, large meetings, and organizational settings. The focus of people is commonly directed towards one individual such as a teacher, presenter, or conductor. A typical EWall application for Centralized Control might leave one person in charge of a Workspace View that is visually accessible to all participants. The person in charge would be responsible for the evaluation and implementation of suggested additions and modifications.

4. Individual and Shared Control

Individual and Shared Control applies to situations in which participants collaborate on a common task yet independently consider issues based on their unique perspectives, interests, foci, areas of expertise, and backgrounds. A typical example is a meeting in which participants use their notepads or portable computers to take notes, to focus in on specific issues, or to temporarily disengage and work on related issues. Individual and Shared Control also allows for the separation between personal and shared information that many people are accustomed to. A possible EWall application for this setting is the joint use of a shared Workspace View on a large display combined with the individual use of personal Workspace Views on portable computers. Participants would copy contents from the shared Workspace View to their personal Workspace Views, arrange and complement the contents in ways that best supports their understanding, and subsequently discuss their conclusions with other participants.

5. Individual and Centralized Control

Individual and Centralized Control is similar to Individual and Shared Control yet assumes one person or a small group of people in charge of coordination, and in control of shared resources and tools. Individual and Centralized Control is typical for home workers and consultants that operate independently, use their own resources and tools, yet act upon explicit instructions by their employers. A possible EWall application for this setting is the collaboration among specialists that use their individual Workspace Views to independently evaluate shared information. Occasionally, a coordinator would ask the specialists to compare and discuss their findings through their Exchange Views and subsequently combine and organize their most essential conclusions on a shared Workspace View.

6. Decentralized Control

Decentralized Control refers to situations in which people work independently on individual or on common tasks. Communication in Decentralized Control settings happens primarily between pairs of people and on a need-only basis. A typical example is the communication among students that call each other over the phone to resolve questions regarding their homework or to coordinate their efforts during joint exercises (peer-to-peer learning and problem solving). Decentralized Control is common among people who are geographically distributed, collaborate asynchronously, work on different tasks, or possess different expertise. The advantage of Decentralized Control is that individuals can simultaneously collaborate on different tasks and that collaborating groups of people can dynamically assemble, merge, separate, and dissolve. The disadvantage of Decentralized Control is that collaborative tasks are difficult to coordinate and that collaborating individuals often do not share a common focus or objective. EWall is particularly suited for Decentralized Control as it helps users to detect overlaps in foci and interests, to search for people with relevant expertise, and to prioritize contributions from large numbers of people.

7. Combined Control

Combined Control refers to situations in which two or more settings converge. Combined Control is typical for the collaboration among independent organizations such as contractors on a construction site. The participants of collaborating organizations usually maintain their existing control structures yet coordinate their activities and share resources and expertise. While the communication between two Centralized Control settings is usually limited to a few individuals, the communication between two Shared Control settings may involve all participants. This means that in a Centralized Control setting the team leaders coordinate the activities of their team members and in a Shared Control setting the team members coordinate themselves. For example, the supervisors of two construction companies could coordinate the activities of their workers or the workers themselves could independently resolve conflicts on site. EWall supports Combined Control by allowing individuals and groups of people to share resources and improve communication by connecting their servers. For example, two research teams may be working on a similar task and consequently decide to connect their Database and Exchange Servers to make all information available to the participants of both research teams.

The dynamic emergence, configuration, combination, and separation of different settings can result in very complex collaborative arrangements. The participants of such arrangements often only know about the clients and servers they are directly connected to, the people they collaborate with, the roles they play, and the tasks they are supposed to perform. To monitor, understand, design, and manage complex collaborative arrangements and work processes may require graphical user interfaces for visualizing and modeling connections among the EWall components of different users, for assigning user roles and work tasks, and for defining access permissions.

Illustration 15 shows different processes, settings, and user roles within the context of a collaborative arrangement. The collaborative arrangement exclusively displays EWall components though participants may be using additional means for communication and information exchange such as email, instant messaging, and video conferencing tools. Furthermore, the collaborative arrangement only reflects a particular moment in time and may dynamically adapt to changing circumstances.

The collaborative arrangement is based on a Noncombatant Evacuation Operation (NEO) Scenario provided by the United States Pacific Command (PACOM) [27]. The scenario illustrates a situation in which a NEO is planned and executed by an Operational Planning Team (OPT) at PACOM. Eight OPT Key Planners that are geographically separated handle the NEO decisions. The Key Planners belong to different PACOM directorates and have access to a large number of OPT Personnel consisting of representatives from special staff and Joint Task Force components as well as five OPT Core Group Members that collect input and post intelligence and assessment information to the OPT web page. The planners and decision makers collaborate in distributed and sometimes asynchronous virtual rooms. (Virtual rooms imply the use of technologies for remote collaboration such as video conferencing systems and application sharing software.) Virtual rooms are developed as necessary to support the needs of the various teams, agencies, and commands throughout planning and execution. Virtual Meeting Rooms are created to conduct situational discussions and briefings as well as to facilitate collaboration among members of the virtual OPT and other participating teams. Virtual Planning Rooms allow PACOM and other members of the interagency working group, along with subordinate and supporting commands, to conduct planning meetings and to monitor progress.

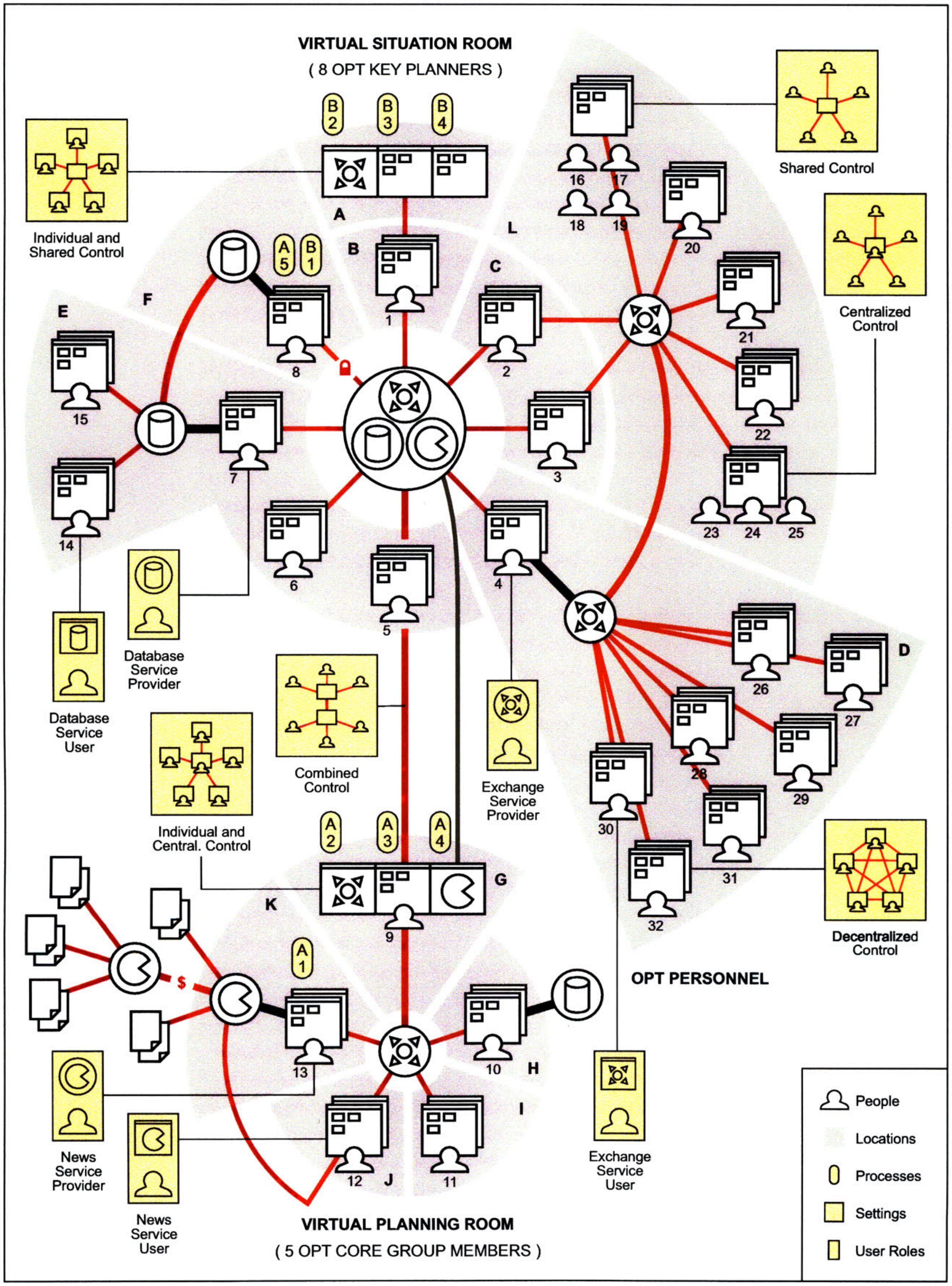


Illustration 15: Examples of EWall processes, settings, and user roles

The EWall proposition for this collaborative arrangement shows the eight OPT Key Planners (Person 1-8) in remote distributed locations (Location A-F). The Key Planners maintain a Virtual Situation Room by sharing a News, Database and Exchange Server as well as an Exchange View and two Workspace Views. The OPT Core Group Members (Person 9-13) also operate from different locations (Location G-K) and maintain a Virtual Planning Room by sharing an Exchange Server. The Exchange Server of the Key Planners and Core Group Members are connected to allow for monitoring and information exchange (Combined Control). The participants also use individual Views and some participants maintain their own servers. For example, person 4 operates an Exchange Server to receive input from OPT Personnel at his organization (Person 26-32). Person 7 shares a Database Server with his two associates (Person 14-15) to automatically merge and archive the contents on their Workspace Views. Person 13 operates a News Server to monitor for task related news on two web sites and a subscription based News Server. Some participants connect to the Servers of communities that have no direct relationship with the OPT. For example, person 2 and 3 connect to an Exchange Server that enables access to several work groups with similar foci and relevant expertise (Person 16-25).

The task of the Core Group Members is to collect information of potential relevance for the Key Planners. Person 9 is in charge of coordinating the efforts of the Core Group Members (Individual and Centralized Control). The Core Group Members use individual Workspace Views to collect information specific to their areas of expertise (e.g. Economy, Geography, Politics, History, and Military). Occasionally the Core Group Members use their Exchange Views to compare and discuss their findings. Person 9 uses his Workspace View to combine and evaluate relevant contents from the individual Workspace Views (A1-3). Person 9 then copies the most essential contents to a News View where it becomes accessible to the Key Planners (A3-5). The Key Planners operate in a similar setting as the Core Group members yet do not assign coordinative responsibilities to one particular individual (Individual and Sheared Control). The Key Planners combine and discuss their conclusions on a shared Workspace View (B1-3). A second Workspace View allows the Key Planners to separate information that is being evaluated from information that has been finalized (B3-4). Some Key Planners operate their personal Servers. For example, person 8 maintains a Database Server that contains information about the names, locations, and availability of people with task relevant expertise. Person 8's Database Server is not accessible to other Key Planners due to the confidentiality of some of its contents.

Implementation Status and Credits for Section A

The Workspace Module functionality introduced in Illustration 8 is implemented. Two News View versions with fixed input feeds as shown in Illustration 9 are operational. The Database Module functionality introduced in Illustration 10 is implemented. The Exchange Module functionality introduced in Illustration 11 is implemented. A standalone version of the Visualization Module as shown in Illustration 13 is operational. The basic setup (red) introduced in Illustration 14a is working. The EWall Application was conceived by Paul Keel and implemented by Michael Kahan, Yao Li, Akshay Patil, Raudel Rodriguez, Mathew Sither, Benjamin Wang, and Patrick Winston. The development of the EWall Application progressed under the consultation of Edith Ackerman, Jeffrey Huang, William Porter, and Patrick Winston.

SECTION B ALGORITHMS

This work proposes a set of computational algorithms for managing computational data (and information) spaces. Interpretation Algorithms supplement data spaces with meaningful relations among data items based on the use and current organization of data items. Transformation Algorithms then utilize these relations in order to help users explore, analyze and exchange the content of data spaces. Interpretation and Transformation Algorithms are divided into four types each:

Types of Interpretation Algorithms:
Level I Algorithms for the recognition of explicit and more obvious relations within spatial arrangements of data items,
Level II Algorithms for the recognition of implicit relations within spatial arrangements of data items,
Level III Algorithms for the recognition of implicit relations based on the collaborative use and history of data items, and
Level IV Algorithms for the recognition of implicit relations based on previously established relations among data items.

Types of Transformation Algorithms:
News Algorithms for the monitoring of modifications to selected data sources,
Database Algorithms for the retrieval of relevant data from selected data sources,
Exchange Algorithms for the prioritized exchange of data between multiple collaborating users, and
Visualization Algorithms for the rearrangement of data items in ways that are inspiring and informative to users.

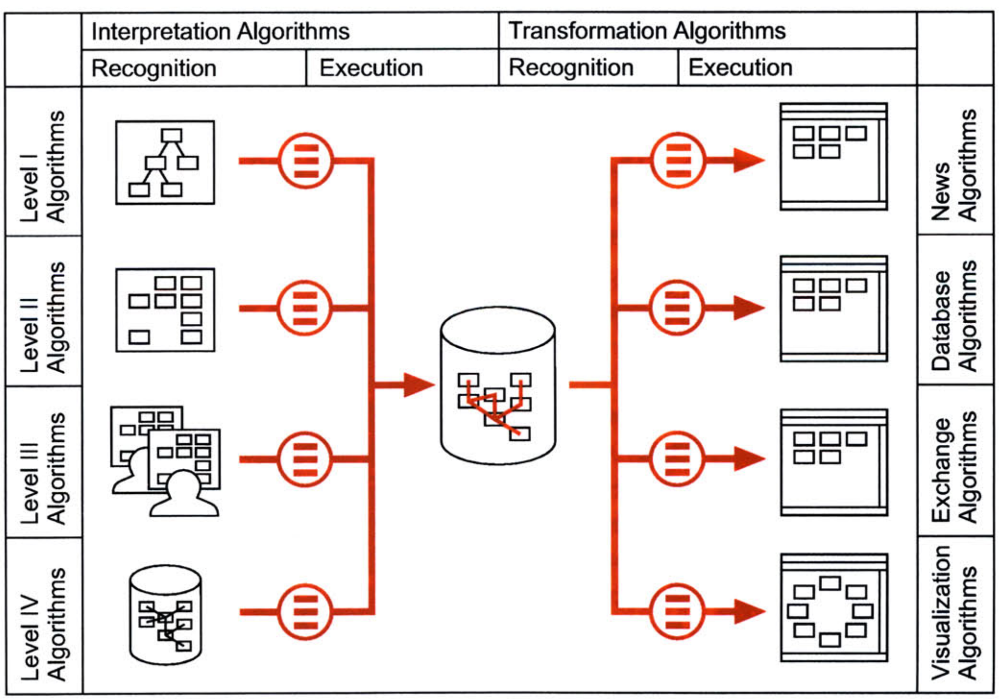


Illustration 1: Types of Interpretation and Transformation Algorithms

Note: This section initially discusses the philosophy and mechanics of a semi-structured data space suitable for the use with the proposed algorithms. Appendix A outlines the distinctions between structured, unstructured and semi-structured data spaces that are essential for understanding the purpose and mechanics of the algorithms. This section then outlines the purpose and mechanics of several selected algorithms that best illustrate the proposed concepts. This work specifically focuses on Level I and II Algorithms as well as Exchange and Visualization Algorithms. The remaining four types of algorithms intersect with well established research areas and are mainly added for completeness as well as to supplement existing concepts with a small set of possibly new and interesting algorithms.

Note: The various observations of human perception and cognition in this work primarily draw from associationist, cognitivist and constructivist theories in psychology, while the computational concepts are commonly portrayed from a connectionist point of view; a distinct area of research within the field of cognitive science. Appendix B provides a brief overview of six different theories in psychology and introduces three distinct research approaches to cognitive science as a means to situate the observations and concepts outlined in this section.

Note: The paragraphs labeled with an **H** reflect on mental models for the construction and manipulation of relations while the paragraphs labeled with a **C** outline computational solutions for the recognition and management of existing as well as the creation of new relations.

Note: This section uses the words Objects, Items, Nodes, and Cards for similar concepts. Objects and Items refer to physical, virtual or mental representations of data, information and knowledge. Nodes refer to joints in networked data, information and knowledge structures. Cards refer to abstract representations of information (see Section A).

Data Space

The proposed algorithms are designed for use with semi-structured data spaces. The use of the algorithms in combination with semi-structure data spaces allows for an information storage and retrieval system that reflects some of the advantages and disadvantages of the human mind.

H: An advantage of the human mind is that its dynamic and self-evolving structure maintains a healthy balance between concrete and ambiguous knowledge allowing for creative interpretations and ideas to emerge. In contrast, a disadvantage of the human mind is that it looses adaptability with age. Established cognitive knowledge never completely disappears but has a diminishing influence relative to persistent and recent knowledge. Hence, knowledge that is continuously reinforced not only persists but also grows stronger and more easily competes with conflicting knowledge. For this reason some pilot schools insist on training only pilots with no previous flight experience. Most of a pilot education is focused on learning procedures. Specific procedures are practiced repeatedly until a pilot follows them perfunctorily. As a consequence, such procedures are most difficult to "unlearn". Pilot schools with high educational standards want to ensure that their procedures are followed precisely. If at all possible, pilots with previous flight experience would require an enormous amount of time to adapt to differing flight procedures. A new procedure would have to be practiced for an amount of time that significantly exceeds the total amount of time it took to acquire the conflicting procedure. Previously acquired knowledge is not only difficult to "unlearn" because it has been reinforced over a long period of time but also because it may have become the basis for subsequently acquired knowledge. For example, if one draws a relation between the concept of a snake and the concept of danger as well as a relation between the concept of danger and the concept of pain then (with some self-reflection) one may assume (and eventually establish) a relation between the concept of a snake and the concept of pain. If later the relation between the concept of a snake and the concept of danger were to prove untrue then not only one but two existing relations would become affected. Due to the complexity of human knowledge structures the detection of an untrue relation may affect an enormous amount of other relations. Consequently, the relations that cause the most damage are the ones that are being established early in life and the ones that are being accessed most frequently. This is why it is very hard for people to adjust to major changes, to accept that their principle beliefs are not true or to simply look at things from a different perspective. In other words, the human mind is, to a great extent, a prisoner of its past. An existing set of relations could only loose its relevance to a competing and more influential set of relations. For example, brainwashing is a commonly known and extremely radical way of imposing competing relations. More subtle approaches include activities such as teaching and advertising.

C: The semi-structured data space for the use with the proposed algorithms operates with data items and relations between data items. Data items contain content, weights, authorship information and a time stamp. Weights are used to indicate the importance of data items or to indicate the number of attached relations. Authorship information is used to independently analyze data items established by different users or to assess the reliability of information. The time-stamp is used to study the chronology and the age of data additions. Relations can be directional and contain weight values as well as authorship information. Directional relations are used to preserve the chronology and dependencies of data items. Weights are used to indicate the importance of relations and to combine multiple overlapping relations between two data items. Authorship information is used to independently analyze relations established by different users and algorithms. Both, data items and relations do not contain any indication about why they have been placed in particular location. This ensures that the data space structure does not impose the views of individual content providers but produces a dynamic network of data items that reflects the combined activities and contributions of all users and operating algorithms. Although not applicable to the theories outlined in this work the annotation of data items and relations is a common approach for the design of data structures. Some researches believe that such annotations are essential to establish and analyze the context of data spaces. For example, Dennis Quan et al. [1] describes a semi-structured data space in which "collections of objects" are created by using predicates to link the collection resource to its elements. Also, Jintai Lee [2] introduced a "decision rational language" to annotate the dependencies among criteria, decisions, and consequences.

The proposed algorithms provide the semi-structured data space with self-structuring capabilities. Users are allowed, though not required, to define relations among data items. The algorithms create relations independent of explicit user instructions, thus encouraging users to focus on collecting rather than relating data items. Unlike users, the algorithms do not construct relations based on the analysis of the content of data items but only on the analysis of user generated spatial arrangements of data items, the collaborative use of data items, and on previously established relations among data items. Consequently, the relations established by the algorithms may be different and less accurate than user established relations. However, considering that the algorithms offer a partially accurate and dynamically evolving data structure independent of human engagement, they do provide a valuable solution for a variety of applications such as the recording and structuring of content produced during brainstorming, decision-making, and problem-solving sessions.

Similar to the human mind, the operations of the proposed algorithms may cause the semi-structured data space to loose its adaptability with age, meaning that early and persistent relations are more likely to dominate over time. Interconnecting multiple smaller decentralized data spaces minimizes this effect. This "society of data spaces" is similar to a society of people (or a society of agencies within the mind of a person [3]) where every individual has a specific focus, only maintains a relationship with a selected group of individuals and may or may not gain influence on the society as a whole. A centralized data space only slowly adapts to change. This is because new relations (that reflect the current activities, focus, needs, and interests of users) do not gain strength fast enough to compete with old and well-established relations. On the other hand, the continuous creation of new data spaces (that slowly integrate with older data spaces over time) allows for a flexible data structure that not only preferences more recent contributions but also reflects contextual changes through the formation of new data spaces.

1. Interpretation Algorithms

Interpretation Algorithms construct relations based on the spatial arrangement, use, and storage of data items. Interpretation Algorithms are separated into four distinct groups: Level I Algorithm recognize explicit relations based on user created spatial information arrangements, Level II Algorithms recognize implicit relations based on user created spatial information arrangements, Level III Algorithms recognize implicit relations based on the collaborative use of information and Level IV Algorithms recognize implicit relations based on previously established relations. All Interpretation Algorithms consist of a Recognition and Execution function. Recognition Functions analyze the input from external information sources and extract the data relevant for a specific Interpretation Algorithm. Execution Functions construct relations based on the data provided by the Recognition Functions.

H: Interpretation Algorithms reflect on and support how humans assimilate new knowledge to existing knowledge as well as how they accommodate and refine their own knowledge structures to new and related situations. In other words, Interpretations Algorithms are modeled after the image of how humans select and organize new data to fit the current expectations and needs, how humans make sense of data received through their five senses, how humans continuously adjust and redefine what they know into new knowledge structures, and how human cognitive processes act in parallel to achieve a particular result.

C: Interpretation Algorithms analyze user activities as a basis for the automatic creation of reasonable relations between data items independent of explicit user instructions. Both, the data items and relations are added automatically to a semi-structured data space whose context emerges and dynamically changes with every new addition. As individual algorithms focus on specific aspects only the combination of multiple algorithms may achieve a particular result. These combinations of algorithms can be modified anytime by adding, removing or changing individual algorithms. Equivalent relations conceived by multiple algorithms are likely to become more influential within the data space. For example, an algorithm may propose a relation between data item A and B because the data items were created by the same user. A second algorithm may support this proposition due to the fact that data item A and B were created during the same time frame. Contradictory relations are not resolved but are added to the data space assuming that at a later point in time the accumulation of additional relations will put an end to the conflict. Consider for example an assembly of objects separated into three groups. Each group contains red and white objects. One algorithm may consider relations among objects of equal color while another algorithm may recognize relations based on the grouping of the objects. The algorithms do not have the capability to determine whether neither, only one or both assumption are true but simply add their concluding relations to the data space with the potential of other algorithms supporting their conclusions in the future. Consequently, <u>the data space is an assembly of both, (conceivably) correct and (conceivably) incorrect relations</u>.

Interpretation Algorithms construct either dynamic or static relations. <u>Dynamic relations</u> (/d) reflect the current state of a system (such as the current state of an information arrangement or a database) and are reevaluated after every event (such as an addition, subtraction, or modification). <u>Static relations</u> (/s) on the other hand are established once and never change or expire but only accumulate over time. Thus, the sum of all static relations represents a system's history or long-term memory while the sum of all dynamic relations represents a system's current state or short-term memory. However, a network structure that evolves through the continuous accumulation of static relations may be dynamic in nature. This means that even though static relations are permanent, the network structure as a whole (context) may be changing. This is because the number of relations between two nodes can increase over time consequently producing varying weights (strength, importance) among the relationships between node pairs. Every weight change may affect the network balance in part or as a whole. Consider for example a relation between node A and B and a relation between node B and C. While the two relationships between the two node pairs are currently of equal importance the later addition of a second relation between node B and C would increase the significance of the relationship between node B and C, thus changing the balance of the network. The continuous accumulation of relations often becomes a highly complex network with a large number of nodes and a very large number of relations. To prevent the accumulation of large numbers of relations, corresponding relations proposed by different Interpretation Algorithms are accumulated. For example: Algorithm 1 suggests a relation between node A and B and a relation between node B and C. Algorithm 2 suggests a relation between node B and C (corresponding) and between node C and D (non-corresponding). The consequent network will connect node A and B with a relation of weight 1, node B and C with a relation of weight 2, and node C and D with a relation of weight 1. The concluding network represents 3 relations with variable weights rather than 4 relations with equal weights. The translation of multiple relations between two nodes into a single relation of higher weight not only reduces the amount of links but also allows for an effective visualization of the network structure by representing relations and weights with lines of varying thicknesses.

The scenario in Illustration 2 demonstrates how multiple simultaneously acting Interpretation Algorithms reflect on user modifications to a spatial information arrangement. The first column (U1-7) displays a user's evolving spatial information arrangement over time. Elements in red color highlight recent modifications. The second and third column (R1-7, E1-7) displays the construction of relations by five independently operating Interpretation Algorithms. The Group Algorithm (gray, dynamic, 1.1.1.) relates cards that the user explicitly grouped with a bounding box. The Link Algorithm (yellow, dynamic, 1.1.3.) relates cards that the user explicitly linked with a rubber line. The Proximity Algorithm (green, dynamic, 1.2.1.) relates cards in close proximity. The Orthogonal Algorithm (blue, dynamic, 1.2.5.) relates cards that align horizontally or vertically. The Addition Algorithm (red, static, 1.2.10.) relates the most recently modified card with the previously modified card (sequence). The last column (T1-7) displays the sum of all established relations between cards with single lines of varying thicknesses. For example, in row 1 the user places two cards onto an empty work space. The Proximity Algorithm (green) establishes a relation between the two cards because the cards are in close proximity (the green circles in R.1 overlap). The Orthogonal Algorithm (blue) establishes a relation between the two cards because they align horizontally (blue line in R.1). The Addition Algorithm (red) establishes a relation between the two cards because they represent the most recent and previous modified card. The sum of the three established relations between the two cards (in E.1) is represented with a single line of thickness 3 (in T.1). The concluding network of relations (in T.7) represents the database structure that can be analyzed and viewed with Transformation Algorithms. The accuracy of the database structure depends on the number, weight and quality of Interpretation Algorithms. A large number of Interpretation Algorithms is more likely to account for the various differing ways in which users spatially relate information. Assigning weights to individual algorithms (or allowing for self-adjusting weights) will ensure that the database can highlight the detection of more obvious relations. The quality determines how easily individual algorithms deal with irregularities and ambiguities when searching for potential relations in spatial information arrangements.

Note: The simple accumulation of relations is sufficient, assuming that the solutions proposed by different Interpretation Algorithms are of equal importance. Additional measures must be taken to deal with Interpretation Algorithms that produce results of varying importance: 1. A first option is to allow users to manually adjust the level of importance for every individual Interpretation Algorithm. This method is time

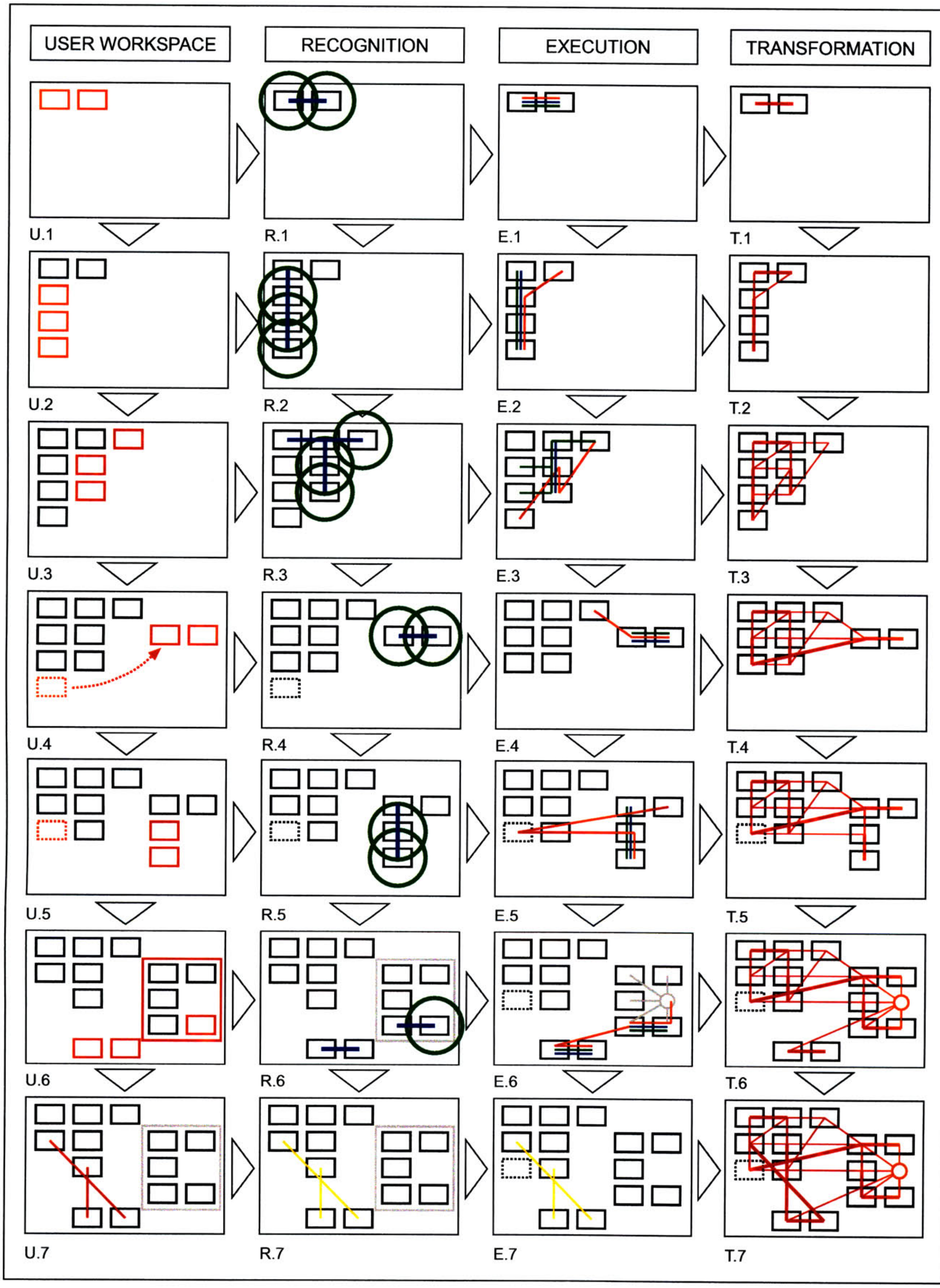


Illustration 2: Scenario of multiple simultaneously acting Interpretation Algorithms

consuming, reflects the opinion of individual users, and instigates a value system that only applies under specific circumstances. 2. A second option is to automatically adjust the level of importance for every individual Interpretation Algorithm through a training program. The training program would have to retrieve the opinions of many users on a large set of test cases. This method is also time consuming yet combines the conclusions of multiple users under varying circumstances. 3. A third option is to automatically adjust the level of importance for every individual Interpretation Algorithm based on indirect user feedback during normal operations. Transformation Algorithms would provide users with information relevant to their current activities. The utilization of such information by individual users would produce positive feedback. This method does not require direct user involvement, reflects the conclusions of multiple users, and is adaptive to changing circumstances. 4. A forth option is to automatically adjust the level of importance for every individual Interpretation Algorithm through a system of computational agents. Every agent would represent one Interpretation Algorithm and negotiate its solutions with agents representing other Interpretation Algorithms. The computational agents themselves would be composed of several self-adjusting algorithms representing "human-like" concepts such as "curiosity", "self-confidence", "trust", "patience", "adaptability", "fatigue", and "satisfaction". Such an agent system would improve through experience, and be adaptive to changing users and circumstances.

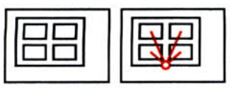
1.1. Level I Algorithms

Level I Algorithms reflect on and support how humans formalize explicit relations in a spatial arrangement of data items. Explicit relations refer to non-ambiguous relations such as relations between data items listed under the same category or data items stored in the same directory. Implicit relations suggest less obvious and often context dependent relations such as relations between data items that are spatially located near each other or data items that contain similar content.

H: Humans commonly choose from a variety of methods for organizing data items. The method of choice usually reflects a number of variables including the time available, the amount of data items to be organized as well as the desired accuracy and flexibility of the data space. Most methods are applicable to both, computationally based data items as well as physical objects. Consider the following examples: Groups separate data items into multiple clusters. Grouping is a very efficient method for an initial separation of data items. The time it takes to add new data items increases with the number of groups. Some computational systems offer Layers for the temporary comparison or combination of multiple groups. Hierarchies are tree-like data structures that allow for the grouping of groups. Hierarchies enable users to organize larger numbers of data items. Categories allow individual data items to belong to more than one group. Categories refer to headings or keywords associated with data items. The time it takes to categorize data items increases with the number of categories. Tables are often used to visualize data items that belong to multiple categories. Networks offer a very flexible but also complex and time-intensive solution for organizing data items. Every data item in a network may be associated with every other data item. The time it takes to accurately establish relations increases exponentially with the number of data items. Networks are commonly used in dynamic and decentralized environments where large numbers of people continuously add, remove and change data items and relations.

C: Level I Algorithms provide users with the capability to easily create explicit relations among spatially arranged data and information items (cards). For example, users can create explicit relations by grouping data items with a bounding box or by linking data items with a rubber line. Level I Algorithms convert these explicit relations into a standardized format that allows for the comparison and merging with other types of relations. While the Level I Algorithms proposed in this chapter are specifically designed for the use with spatially arranged data items, they can be conceived for other types of arrangements as well. For example, Level I Algorithms might identify and convert relations implied in the organization of electronic file systems or the location of (electronically tagged) books in libraries. Unlike Level II, III, and IV Algorithms, Level I Algorithms neither assume relations among data items nor approximate the value and validity of relations.

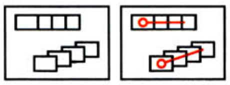
1.1.1. Group /d



H: Typical decision-making, brainstorming and design processes commonly start with the collection and organization of relevant data or information. This first phase often accumulates a lot of information that must remain easily accessible and comparable. The grouping of information provides a fast and easy way to get people started and encourages people to carefully consider the similarities and differences among collected pieces of information. Throughout this initial investigation people develop a good understanding about the problem space and the range of information available. The grouping of information becomes less advantageous as the amount of information grows hence other organizational methods will eventually become more beneficial.

C: The Group Algorithm provides users with the functionality to group cards. The Group Algorithm then automatically constructs relations between these grouped cards. Contemporary software applications provide a diverse range of functionality for the grouping of data items. For example, typical drawing applications allow users to group individual drawing elements so they can be manipulated as a unit. Drawing applications also allow users to assign drawing elements to different layers to allow users to independently operate and compare groups of drawing elements. In comparison, the Group Algorithm does not require the user to explicitly group cards but simply assumes that if card B is positioned within the spatial boundaries of card A then card A is a group with card B as its member. If card C is positioned inside card B then a hierarchy emerges with A as the root, B as a branch and C as a leaf. If the user moves card A then card B and C are being moved accordingly. If on the other hand the user moves card B then only card C follows and card A maintains its position. To preserve the hierarchical structure, the Group Algorithm converts all branches into directional relations concluding the previous example with one directional relation from A to B and one directional relation from B to C.

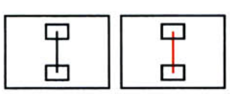
1.1.2. Stack /d



H: Spatial information arrangements are most effective if all data items remain visually accessible. A person's visual range increases through eye, head and body movement. Eye movement covers a visual range not much bigger than a computer display. Head movement increases the visual range significantly. Body movements such as turning, bending and walking allow access to virtually unlimited sizes of information arrangements. While the human visual range seems easily expandable, the human ability to mentally comprehend and deal with large amounts of information is not. Information displays bigger than what a human can mentally comprehend may not prove effective. Contemporary information displays remain significantly smaller than what humans can deal with. Consequently, people are not yet confronted with the problem of dealing with large information spaces but with putting large amounts of information onto small information spaces. People commonly deal with this problem by overlapping information items. For example, operating systems allow users to overlap individual application windows on their desktops. The visible elements of partially hidden application windows not only remind users of what applications and files are currently open but also hint at some of the content. Game cards provide another example. Game cards usually contain the core content in the center while providing hints about the content along the card border. This allows players to organize and overview multiple overlapping game cards. An arrangement of partially overlapping application windows or game cards is more meaningful to the people that created the arrangement and to people that have knowledge about the hidden content.

C: The Stack Algorithm provides users with the functionality to move cards to the front or to back of other cards. The Stack Algorithm then automatically constructs relations between these overlapping cards. Relations are established if a sequence of cards is detected. A sequence of cards is defined as a set of three or more cards that partially overlap, whose top left edges or center points line up and are evenly distributed. The first card of a sequence is the card that is in front of all other cards. The Stack Algorithm constructs a directional relation from the first card to the second card, from the second card to the third card, and so on. If the user moves the first card of a stack all other cards are moved accordingly.

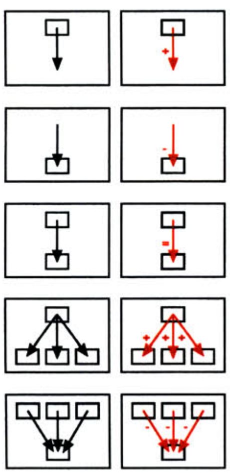
1.1.3. Link /d



H: A natural way of establishing relations between data items is to draw lines between related data items. Various software applications such as Visio [4], SmartDraw [5] and Inspiration [6] provide sophisticated functionality for connecting data items with lines. The lines can be non-directional (no arrow heads), unidirectional (one arrow head) or bidirectional (two arrow heads). Using lines is not a convenient way of establishing relations as users have to continuously ensure 1) that there is enough space between information items to clearly see the connecting lines, 2) that lines don't intersect with other lines or data items, and 3) that the amount of lines remains visually manageable. Hence, due to the effort involved, lines are less often used for establishing relations than for visualizing processes, networks or relations among small numbers of data items. Some software applications are capable of automatically rearranging data items to minimize the amount of overlapping lines and data items [7] [8]. However, the constantly changing locations of data items is likely to interrupt the work of users and may prove less effective in situations where users have to quickly and effectively establish relations among a large number of data items.

C: The Link Algorithms provides users with the functionality to connect data items with rubber-lines. The rubber-lines can contain arrows and can be displayed in front or behind cards. While this research does not consider this functionality as an effective approach for relating data items it is only offered for the use in rare occasions as well as for the testing and adjusting of algorithms. The Link Algorithm does not change user-established relations but only saves them in a format that allows for the comparison with relations established by other algorithms.

1.1.4. Progress /s



H: In some cases it may prove beneficial to carefully keep track of the development of a decision-making session. The data collected may be used to monitor the development of a decision-making session, to return to an earlier state of a decision-making session, to analyze the performance of a decision-making session, or to reconstruct and evaluate a decision-making session.

Essential for the tracing of a decision-making session is the concise recording of the times, contents and shifts in focus of all the issues considered. Imagine the following example: A group of decision-makers first discusses issue 1, then moves on to issue 2, and finally converges on issue 3. The discussion about issue 3 reflects and builds on the discussion about issue 1 and 2 hence the focus of the conversation goes back and forth between the current and the previous two issues. A successful tracing of this process would not only record the chronological order of issues (1-2-3) and their contents but also include the chronological order of shifts in focus (e.g. 1-2-1-3-2-3). Shifts in focus may be recognized based on the content of a discussion, based on the hand gestures and eye movements of people in regards to visual materials, or based on the changing location of visual materials. However, the continuous shifts in focus within human minds remain inaccessible to observers. Furthermore, multiple collaborating users may not simultaneously focus on the same issue. Consequently, it is not feasible to accurately track but only estimate the focus of one or more people during a decision-making session. However, even a partially accurate tracing of shifts in focus is often helpful in gaining a deeper understanding of the decision-making process. Optimally, the tracing is done by an observer as opposed to the people concentrating on the decision-making process. However, the quality of the data collected may improve significantly through explicit commentaries by the decision-makers.

Of potential interest to observers and analysts of decision-making sessions is the decision-makers evolving situated understanding and knowledge of a given problem space. This requires observers to carefully keep track of the emergence of new issues as well as the termination, combination, separation and refinement of existing issues. A possible mapping of such events has been demonstrated during a test with two student volunteers and one observer [9]. The two volunteers were asked to develop an outline for a research paper in their area of expertise. While during the first phase the volunteers were mainly focused on collecting and evaluating content, later stages tentatively started to concentrate on relating and organizing content. The observer kept track of the development by citing (and chronologically numbering) every new item under discussion (in a random location) on a commonly accessible white board. The volunteers were asked to inform the observer whenever a new item was related to a previous item (e.g. if a new item was inspired by a previous item or if one or more new items substituted one or more previous items). The observer would record such relations by drawing lines between items on the white board.

Illustration 3b visualizes the conclusions of this experiment. For the purpose of this analysis, the items listed on the white board are represented as small rectangular boxes and arranged in chronological order. Because the content associated with individual items is not relevant for this analysis the boxes only display sequential numbers. The frequent horizontal displacement of boxes is to increase the readability of links. The links in black color indicate relations between consecutive items (e.g. item 2 concludes from item 1; or item 7, 8, and 9 conclude from item 6) while the links in red color hint more significant shifts in focus (e.g. item 44 concludes from item 33 and 42). The characters (S, V, C, A, T) indicate the occurrence of a Substitution, Variation, Combination, Addition or Termination. A Substitution (S) indicates that a new item concludes or replaces a previous item. For example, a new item may suggest the use of "wood" because the previous item concluded that "wood is less expensive than steel". A Substitution is established if an item maintains one link to one older item. A Variation (V) indicates that multiple new items conclude or replace one previous item. This may happen if for example a book chapter is divided into multiple sub chapters. A Variation is established if multiple items link to one older item. A Combination (C) indicates that a new item concludes or replaces multiple previous items. This may happen if a group of items is combined because of their contextual overlaps or redundancies. A Combination is established if one item maintains links to multiple older items. An Addition (A) indicates that a new item was created uninfluenced by any previous items. An Addition is established if an item does not maintain links to any older items. A Termination (T) is established if an item does not maintain links with any consecutive items. A termination suggests that an item may haven been forgotten or that an item is no longer relevant.

Illustration 3a provides an abstract visualization of several Substitutions, Variations, Combinations, Additions and Terminations within the context of two distinct types of decision-making methods. A dot represents an individual data item such as a comment, an idea, a possibility or a solution. A row of dots represents the total of all active data items within a specific time period. The red lines display the evolution of dots and consequently also illustrate cases of Substitutions, Variations, Combinations, Additions and Terminations. The top half of the graphic illustrates an analysis-based method that progresses from a narrow to a more complex data space. This method is usually more effective during the initial stages of a decision-making process and often produces a lot of data within a short period of time. The explorative nature of this method suggests a tendency towards collecting rather than organizing data. Consequently, Additions and Variations are more common for analysis-based methods. The bottom-half of the graphic illustrates a synthesis-based method that is directed

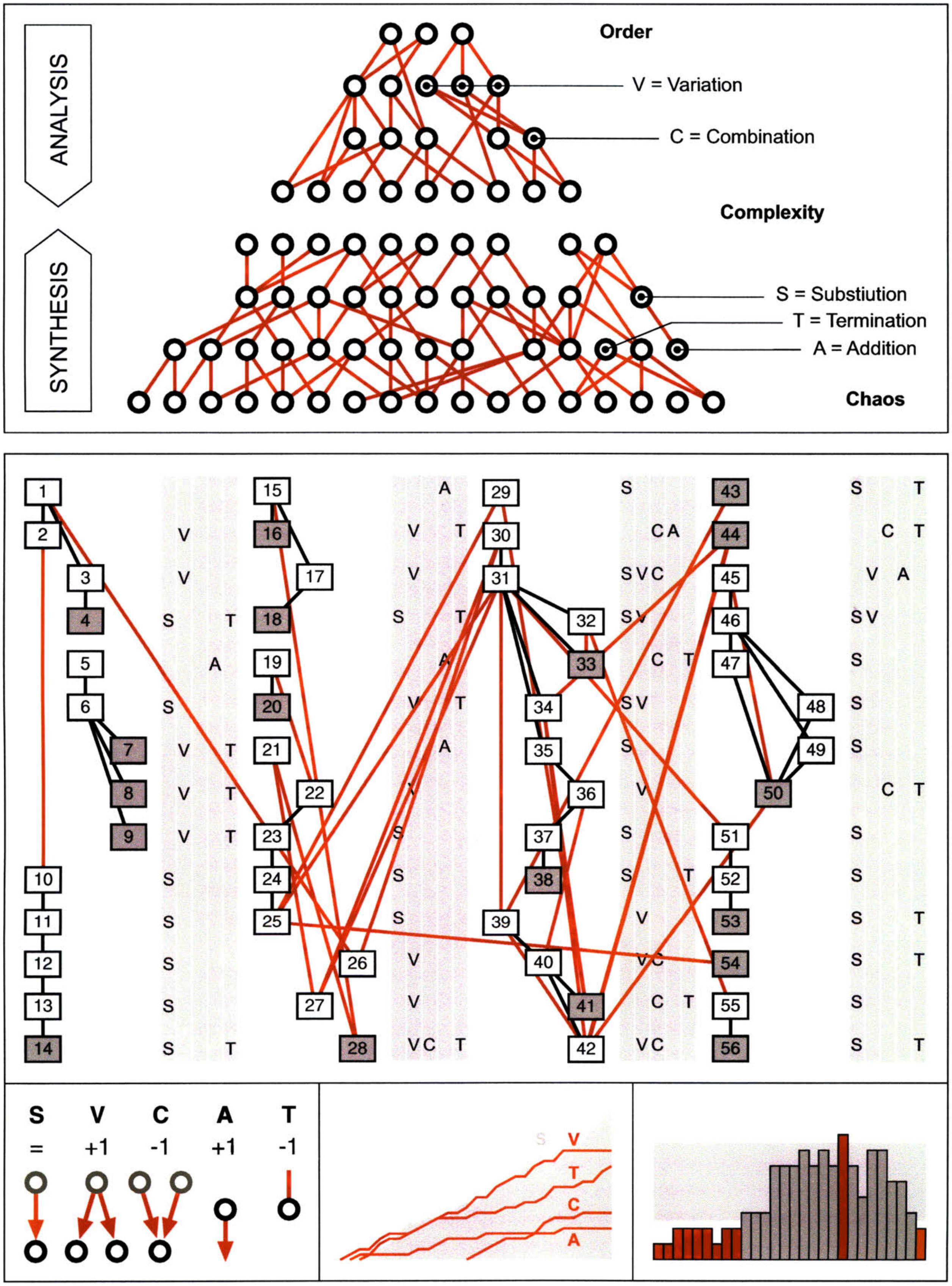


Illustration 3a-e: Experiment illustrating the concepts of Substitution, Variation, Combination, Addition, and Termination.

towards the simplification of complex data spaces. Because a decision-making session usually starts out with a very narrow data space, synthesis-based methods become more meaningful during the later stages of decision-making sessions. Terminations and Combinations are more common for synthesis-based methods. Optimally, analysis- and synthesis-based methods coexist during decision-making sessions. The continuous alternation between collecting and organizing data is more likely to produce and maintain a mentally manageable data space. The combined use of the two methods also ensures a healthy balance between the creative exploration for new and the careful association of existing data items. The distinction between the analysis- and synthesis-based method is similar to Indurkhya's [10] distinction between similarity-based and similarity-creating metaphors as well as to Nueckles's and Janetzko's [11] distinction between analysis-based and synthesis-based processing.

Illustration 3c-e show the conclusions of an experimental analysis of the data displayed in Illustration 3b. The purpose of this analysis was to investigate shifts in focus between the accumulation of new and the association of existing data items.

Illustration 3c introduces a counting system for the occurrences of Substitutions, Variations, Combinations, Additions and Terminations. Variations and Additions increase the amount of active data items by one point while Combinations and Terminations decrease the amount of active data items by one point. Substitutions leave the amount of active data items unchanged.

Illustration 3d displays the total amount of Substitutions, Variations, Combinations, Additions and Terminations (y-axis from 0 to 30) after every addition (x-axis from 0 to 56). The linear development of the Substitution graph suggests that older items almost always influenced the development of new items. Unlike the Substitution graph, the liner development of the Variation graph suddenly stops progressing close to the end of the decision-making session. This may be because the contextual granularity of items did not allow for additional separations or because the focus shifted on concluding the decision-making session. The Combination graph only displays activity during the second half of the decision-making session while the Addition graph displays activity almost solely during the first half of the decision-making session. This is because the initial stages of decision-making sessions more commonly focus on collecting rather than combining items. In other words, the number of potentially useful item combinations increases and the need for new items decreases with the growing amount of items. The Termination graph becomes more active towards the end of the decision-making session. This is due to the forced breakup of the decision-making session that leaves multiple threads of thoughts undeveloped (see item 50, 53, 54 and 56).

Illustration 3e displays the total of all Variation, Combination, Addition and Termination values (y-axis from 0 to 10) after every second addition (x-axis from 0 to 56). The total is calculated by subtracting the sum of all Combination and Termination values from the sum of all Variation and Addition values. Consequently, short bars represent a structured data space while long bars represent an unstructured data space. In other words, the presence of short bars tentatively propose to accumulate new data items while long bars suggest the potential need for associating existing data items. The gray area indicates a healthy balance between the associations of existing versus the accumulation of new data items. Optimally, the length of the bars would constantly alternate but stay within the limits of the gray area. The previously explained analysis-based method is likely to produce ascending bars while the application of the synthesis-based method should foster descending bars. Because the initial stages of a decision-making session are usually devoted to exploration, the bars are expected to ascend during the first phase and descend during the second phase of the decision-making session. The bars in Illustration 3e however remain low for the first third of the decision-making session before starting to reflect the expected behavior. The following observations from Illustration 3b provide a possible explanation: The hierarchical structure connecting item 1 - 9 indicates that the volunteers had initially a very good idea about the main trust of their paper. However, the first segment of this initial structure terminated with items 7 - 9 and the second segment ended, after multiple substitutions, with item 14. The buildup of a third segment starting with item 15 also concluded soon after with item 18. The consequent detachment of items 1 - 18 from the reaming network suggests either that the volunteers dismissed or suspended their initial ideas or that the volunteers quickly concluded a first set of independent issues before focusing on the main task. The multiple terminating threads between item 1 and 18 prevented a constructive buildup of relations and consequently postponed the ascending of bars in Illustration 3e. (Low and non-ascending bars do not suggest a problem but only an unusual situation that may require attention.) Starting with item 19, a more solid network of relations emerged which finally produced the expected bar lengths. The volunteers' continuous efforts to establish relations among data items prevented the bars from exceeding the suggested maximum length. The descending bars toward the end of the decision-making session suggests that the volunteers successfully managed to minimize the complexity of the data space, that the volunteers managed to converge towards a solution or that none of the data items provoked any further consideration.

While the usefulness of this experimental analysis is yet to be proven it does suggests a possible approach for controlling and visualizing the balance between accumulating new and associating existing data items.

C: The Progress Algorithm provides users with the functionality to balance the accumulation of new and the association of existing cards. For every new card, the user is supposed to draw a rubber-line between the new card and all previously created cards that inspired the creation of the new card. The rubber-lines are invisible by default but can also be rendered in a light gray. The Progress Algorithm adjusts the progress weights of every individual card based on the previously outlined analysis. A positive progress weight is assigned to cards that represent a Variation or an Addition, a negative progress weight is assigned to cards that represent a Combination or a Termination, and no progress weight is assigned to cards that represent a Substitution. The user can view the total of all progress weights in a bar diagram similar to the one shown in Illustration 3e. The rubber-lines and the progress weights can also be combined with the relations and card weights generated by other algorithms. Optimally, the rubber-lines are drawn by an observer rather than by the decision-makers themselves.

1.2. Level II Algorithms

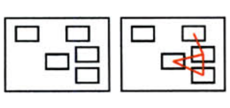
Level II Algorithms reflect on and support how humans create and interpret implicit relations from spatial information arrangements.

H: Humans commonly organize and comprehend information by spatially arranging individual information items. The information items are commonly arranged in clusters, rows, columns, or tables. Examples range from simple information arrangements on refrigerators in private households to very complex information arrangements on shared displays in command and control centers. While some information arrangements are specifically developed for accessing information, other information arrangements are used to discover relations among information items. William Pena's Architectural Programming method [12] provides a good example of how large groups of people can take advantage of such information arrangements during collaborative decision-making sessions. For every issue that is discussed, every piece of information that is collected and every idea that is proposed, the collaborators pin up a card that serves as a reminder of what has been said or discovered. This card arrangement serves as a group memory representing the shared understanding of all collaborating users. The static location of cards not only allows users to easily navigate the card space but also serves to preserve implicit relations among cards (e.g. cards located near each other). The sum of all relations among cards (or context of a card arrangement) is a combination of the card contents, the spatial location of cards, the sequence of card additions, the information exchange between collaborating users during the development of a card arrangement, as well as the knowledge and background of individual users. Hence, essential contextual information of spatial information arrangements exists partially within the minds of the creators of an information arrangement. In conclusion, William Pena's methodology highlights two essential actives to reduce the human cognitive load. First, the externalization of human knowledge into cards provides effective means to easily collect and share knowledge. Second, the spatial card arrangements allow humans to more easily comprehend, organize and analyze information.

Note: To understand how humans construct and perceive relations among spatially arranged objects has been the subject of various research projects mainly within the fields of Cognitive Science and Artificial Intelligence. The various phenomena that allow humans to deal with this kind of perceptual organization (including the recognition of relations between parts and wholes [13]) are often referred to as "Gestalt Perception" [14]. While Cognitive Science pursues these issues as a basis to learn about human cognitive processes and loads, Artificial Intelligence commonly investigates this topic as an inspirational foundation for research on computer vision. Even though most of the research in this area is focused on how to recognize and relate objects in space rather than to determine relations between spatially arranged information fragments, the research findings provide valuable clues to both investigations. For example, David Kirsch [15] tested the human cognitive abilities on a spatial arrangement of coins. Human subjects were asked to determine the total value of all coins with and without rearranging the coins. Kirsch noticed that the time needed and the mistakes made increased significantly if the subjects were not allowed to spatially rearrange the coins. Kirsch concluded that "humans amplify their cognitive abilities by adapting their environments of action to environments where they can get the best result from their limited cognitive resources". This study supports the theory that through the manipulation of spatial information arrangements the human cognitive load is greatly reduced allowing for a better understanding of large and complex information spaces. While computer systems are capable of retrieving (and consequently processing) the contents of human created spatial information arrangements (such as the value, sizes and locations of coins) they are not yet able to recognize implicit relations among individual information items (such as the grouping of coins by value). Research on how humans construct and perceive such relations will allow for the development of computational mechanism with similar capabilities. For example, Max Wertheimer's [16] research on perceptual organization provides valuable insights into how humans perceive relations among spatially arranged objects by suggesting a set of principles for detecting object relations based on concepts such as size, proximity, similarity, or continuation. Wertheimer explains that the sequence "xx xx xx xx" is more likely to be read as ab/cd/ef/gh rather than a/bc/de/fg/h. On the other hand, the sequence "xX Xx xX Xx" is read either as ab/cd/ef/gh or a/bc/de/fg/h. Wertheimer also offers interesting conclusions from cases in which differing concepts support or contradict each other. Wertheimer's work not only inspires but also reveals various opportunities for additional investigations within the area of perceptual organization. The discussion on Level II Algorithms below provides complementary thoughts on the issues of perceptual organization and suggests computational mechanisms capable of simulating such cognitive processes.

C: Level II Algorithms are designed to discover implicit relations in user created spatial information arrangements. Every Level II Algorithm focuses on one particular criterion such as the horizontal or vertical alignment of cards. The quality of the result depends on what combination of algorithms is applied to what kind of information arrangement. For example, people that write from top to bottom (as in some Asian countries) are more likely to organize cards vertically. Consequently, an algorithm focusing on vertically aligned cards may prove more effective than an algorithms focusing on horizontally aligned cards. While Level II Algorithms cannot inclusively portray the context of an information arrangement, they may, with increasing sophistication, offer solutions similar to a human observer. Because these algorithms do neither change nor obstruct the work process of users, their conclusions can be seen as a complimentary product for users to easily search, merge and analyze information arrangements.

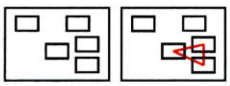
1.2.1. Proximity /d



H: Humans often recognize relations between objects based on spatial proximity. For example, relations may be assumed between books next to each other on shelves or documents stored in the same folder. Similarly, humans also take proximity into consideration when organizing or arranging information. For example, a pin board might show job advertisements on the left, announcements in the center and events on the right. Proximity is a good indicator for the potential existence of relations that often outbids other competing concepts.

C: The Proximity Algorithm establishes relations between cards based on spatial proximity. This algorithm considers two cards in close proximity if the distance between the two cards is equal or greater than 0 and less than the diagonal size of the smaller card. The weight of every relation varies based on the distance between the two cards. No spacing between two cards produces a relation of weight 1 while a distance equal the diagonal size of the smaller card will produce a weight of 0. The impact of a "proximity" relation between two cards depends on its weight and the potential existence of other types of relations between the two cards.

1.2.2. Cluster /d

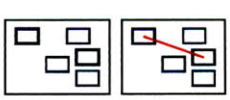


H: Humans recognize clusters of objects based on the varying distances between individual objects. For multiple stacks of documents it is easy to assume that documents from the same stack have more in common than documents from different stacks. While such a conclusion may appear natural to humans, computational systems are not able to recognize such relational hints in the spatial clustering of objects. For example, an operating system does not notice a correlation between a set of files clearly grouped on a computer desktop. Only if the files are located in the same directory or contain similar content then certain computer applications are able to establish relations among these files.

C: The Cluster Algorithm recognizes clusters of spatially arranged cards. A cluster is defined as an assembly of cards that is spatially detached from other cards or assemblies of cards. The Cluster Algorithm first searches for a set of cards in which every card is in close proximity to every other card (see 1.2.1. Proximity Algorithm). The Cluster Algorithm then keeps adding cards whose distance from previously collected cards is less then the longest distance between any of the previously collected cards. The same procedure is recursively applied to all cards that have not yet become part of a cluster. The Cluster Algorithm concludes the process by constructing relations among all cards in a cluster.

Note: The detection of clusters of objects has been the subject of various research projects. Sheel Dhande [17] separates research in this area into three categories: 1. Recognition of Gestalt properties: such as the work by Albert Zobrist and William Thompson [18] on detecting curvilinear continuity, good closure, and overall goodness grouping. 2. Segmentation and object detection: such as the work by David Lowe [19] on visual grouping for object recognition as well as the work by Rakesh Mohan and Ramakant Nevatia [20] on segmenting scenes into objects and their components. 3. Modeling general systems: such as the work by Arnon Amir and Michael Lindenbaum [21] on generic grouping algorithms as well as the work by Eric Saund et al. [22] on sketch recognition.

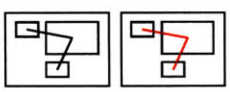
1.2.3. Similarity /d



H: The visual similarities among objects represent another influential factor when establishing relations between objects that can be both, supportive or distracting. It is often tempting to assume a relation between two similar looking objects before considering other relational aspects. The relational appeal of similar objects depends on the level of similarity as well as the distance between the similar objects. Imagine the following example: A movie listing shows the first movie in red color on line 1, the second movie in green color on line 2 and the third movie in yellow color on line 3. The same color scheme then repeats for all remaining movies. While some people would browse this listing in sequential order from top to bottom, others would view the listing color by color. A small experiment with a few people has confirmed this hypothesis. Interestingly, one person even missed the movie he was looking for because he presumed that every individual movie listing is made up of three lines with the movie title displayed in red. Consequently, this person would only scan the lines in red color assuming the other lines only contain actor and scheduling information.

C: The Similarity Algorithm establishes relations between cards of similar appearance. Similarity is assumed if (except for similarities caused through default settings) two cards use the same background color, the same font and font size, the same picture or the same heading. The weight of a "similarity" relation is based on the distance between two similar cards. A weight value of 1 is assigned to a "similarity" relation that connects two cards that are within a distance closer than 5 times the average card size. The weight value then decreases linearly with increasing proximity. No similarity is assumed for distances greater than 15 times the average card size. This formula ensures that only relations that connect cards located within a reasonable visual range are considered for the comparison with competing relations established by other algorithms.

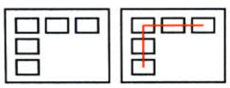
1.2.4. Exception /d



H: While humans are good at recognizing similarities between objects they are even more sensitive to exceptions. The advertising industry very much depends on this fact. An advertisement that is distinct from its environment and other advertisements has a better chance of arousing the interest of potential customers. Essential components for advertisement posters not only depend on eye catching pictures, easy readable fonts and expressive slogans but on the format of the poster and how the poster visually distinguishes itself from its immediate environment. For example, a small number of rectangular sheets among many square sheets are likely viewed first. Similarly, a small number of red sheets among many green sheets are also likely to attract the initial attention of viewers. Humans do not consider multiple exceptions in parallel. Research has shown that if a volunteer is asked to find all the green T's in a field of multi-colored characters they are likely to explore the search space for green characters and T's separately [23]. This means in regards to the previous example that a human is more likely to first compare all sheets that are red or all the sheets that are rectangular rather that all the sheets that are both, red and rectangular. Humans recognize exceptions not only due to dissimilarities among objects but also due to the unexpected appearance of objects such as for example objects that have not been noticed previously, objects that appear suddenly, or objects that are located in odd or wrong locations. The human ability to distinguish between familiar and novel stimuli as well as the human tendency to devote more attention to novel stimuli is known as "habituation" [24].

C: The Exception Algorithm recognizes cards that distinguish themselves from other cards through size, location, background color, font type, font size or picture properties. A card is considered exceptional if it is at least twice the size of the next smaller card, if the distance to the closest card is at least twice as big as the average card distance, if no other card is using the same background color, font type or font size, or if the picture contains a color intensity (the sum of the hue and saturation values of all pixels in a picture) that is at least twice as high as in any else picture. The Exception Algorithm constructs relations between an "exceptional" card and every card directly related (relations established by other algorithms) to it. Consequently, the Exception Algorithm does not change the existing structure of relations but only complements (strengthens) existing relations. This ensures that not only the (visually accessible) spatial arrangement but also the concluding (computationally accessible) relational structure reflects the significance of "exceptional" cards.

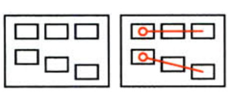
1.2.5. Orthogonal /d



H: Relations can also be assumed among objects that align horizontally or vertically even if the objects are not located near each other. Imagine a table that lists resellers in horizontal and product names in vertical direction. The cells that intersect reseller rows and product columns display the amount of sold items or remain empty if a reseller does not carry the product. The horizontal and vertical relations among cells can be recognized easily even if only a few cells contain values. However, since the proximity among objects visually dominates, the relations among orthogonally aligned objects are more likely recognized if the objects are located near each other.

C: The Orthogonal Algorithm establishes relations among cards that ling up horizontally or vertically. The variance among the upper (or left) card borders must not be more than 1/10 of the average card height (or width). A non-directional relation is added between each pair of neighboring orthogonally aligned cards. The Orthogonal Algorithm is designed to be used in combination with the Proximity Algorithm. The combination of these two algorithms ensures that relations between orthogonally aligned cards in close proximity outweigh relations between orthogonally aligned cards that are far apart.

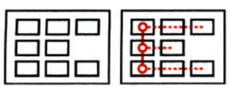
1.2.6. Sequence /d



H: A sequence contains a series of objects distributed along a straight or curved line such as a bunch of game cards in a player's hand. The order of objects in a sequence may be intentional or random. Both the content of the objects as well as the order of the objects can provide clues about the relations among the objects. A sequence usually contains a small number of objects that can be easily compared and rearranged. Analogous to proximity based arrangements, sequence based arrangements are more commonly used for an initial or preliminary grouping of objects. While proximity-based arrangements require less planning, sequence based arrangements are visually easier to comprehend. Sequences have a tendency to visually dominate other types of arrangements. For example, experiments have shown that children under the age of 5 fail to notice quantities among sequences of objects. Children at this age identify longer sequences of objects as the sequences containing the most objects even if these sequences contain same many or less objects [25]. Also important to notice is that sequence-based arrangements are built and read differently by different cultures. For example, western cultures usually build and read sequences from left to right and from top to bottom. Western cultures also have a tendency to favor rows over columns. Some researchers even suggest that such cultural reading habits might affect a wide variety of processes underlying visual perception and thinking [26].

C: The Sequence Algorithm establishes relations among a series of at least three cards of similar size that are equally distributed along a straight line. The rules demand 1) that the variance among the upper (or left) card borders must not be more than 1/10 of the average card height (or width), 2) that the difference between the smallest and biggest card height (and width) must not vary more than 1/5 of the average card height (and width), and 3) that the shortest and longest distance between the top left edges between any two adjacent cards must not vary by more than 1/5 of the average distance. A card sequence is connected with directional relations meaning that the first card is pointing to the second card, the second card to the third card and so on. The first card is assumed to be the top- or left-most card of a sequence.

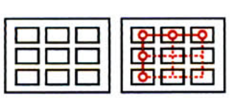
1.2.7. List /d



H: A list can be seen as a sequence of sequences. More specifically, a list is a collection of multiple sequences aligned in horizontal or vertical direction. Imaging three rows of boxes; the first row contains boxes with big-size screws, the second row contains boxes with medium-size screws, and the third row contains boxes with small-size screws. While the rows represent sequences of boxes of screws of similar size, the whole assembly represents a list of three sequences. Lists are one-dimensional meaning that (unlike tables) primary object relations are found either in horizontal or vertical direction (not both). In regards to the previous example this means that the boxes in a row are directly related (similar sizes of screws) while boxes in a column are only indirectly related (screws).

C: The List Algorithm establishes relations among multiple sequences of cards. The procedure is essentially the same as for the Sequence Algorithm. The List Algorithm encloses all cards of a sequence with a virtual bounding box. The List Algorithm then processes the bounding boxes in the same way the Sequence Algorithm processes individual cards. The List Algorithm only creates relations among the lead cards (top- or left-most cards) of sequences. Because a list is structurally and visually more significant than its sequences, the default weights of relations created by the List Algorithm are set to be twice as high as the once created by the Sequence Algorithm.

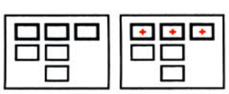
1.2.8. Table /d



H: A table is an arrangement of objects in which each object simultaneously belongs to one horizontal and one vertical sequence. Tables are commonly used for two-dimensional data sets. For example, the rows of a table could display addresses while the columns could visually align individual address components such as names and telephone numbers. In this example, the rows are of contextual significance highlighting relations between names and phone numbers while the columns are only of structural importance indicating some sort of communality among all the names and among all the phone numbers. Sometimes sequences occupy more than one row or column. For example, individual addresses could occupy two rows displaying the name and phone number above the postal address. The individual addresses could be visually separated by increasing the spacing between addresses or by indenting the rows displaying the postal address. In conclusion, the key for the correct interpretation of a table requires both, the visual identification of horizontal and vertical sequences as well as an inquiry into whether these sequences are of contextual or structural nature.

C: The Table Algorithm recognizes and establishes relations between horizontally and vertically intersecting lists. Two lists are considered to be intersecting if one horizontally oriented and one vertically oriented list share at least 90% of the cards. The Table Algorithm only creates one relation between the lead cards (top- or left-most cards) of both lists. Because a table is structurally and visually more significant than its lists, the default weights of relations created by the Table Algorithm are set to be twice as high as the once created by the List Algorithm (hence four times as high as the once created by the Sequence Algorithm). The Table Algorithm uses its own version of the Sequence Algorithm allowing it to recognize "incomplete" sequences. An incomplete sequence is a sequence with one or more missing cards. Hence, the shortest and longest distance between the top left edges between any two adjacent cards may exceed 1/5 of the average distance if the gap between the two adjacent cards accounts for one or more potentially missing cards. This functionality is essential for the recognition of tables because, unlike sequences and lists, tables usually do not receive additions in sequential order and only seldom display complete data sets (e.g. some names in an address list may not be associated with a phone number).

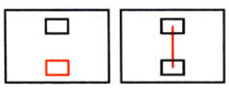
1.2.9. Heading /d



H: Headings are short sentences or combination of keywords that do not add content but help to structure content. Headings are commonly used in text documents to introduce chapters. These headings indicate with a very few words what each chapter is about. Even though the text content is independent of its headings, headings can help readers 1) to understand directional shifts between chapters, 2) to more easily search and navigate a text document, and 3) to recognize communalities among different issues raised in a chapter. Headings are also used in combination with sequences, lists and tables and commonly located next to the top- or left-most object of a sequence, along the top row or left column of a list, or along the top row and left column of a table. Mostly, headings can be identified through their unique appearance such as large font size or distinct color. Object relations emphasized by headings should and often do dominate object relations implied through the spatial arrangement of objects. The summarization of communalities between multiple objects with only a few words is an essential human quality that allows people to effectively communicate structural information about objects arrangements. For example, the skillful labeling of file cabinets, drawers and folders provides helpful hints about the location of specific files.

C: The Heading Algorithm detects and increases the weights of cards that are used as headings. The potential presence of such heading cards is assumed if at least two cards share two or more of the following features: 1. A card uses a font size bigger than the average font size, 2. a card uses only bold characters, 3. a card uses a distinct font or background color or 4. a card acts as the lead card of a sequence, a list or a table. Even though this method may not guarantee the accurate detection of all heading cards it can effectively locate the more obvious heading cards without the need for a semantic analysis of the card contents. This version of the Heading Algorithm is based on the assumption that users create heading cards in combination with sequences, lists and tables while using the functionality offered by the Group Algorithm for the labeling of less structured card assemblies. The Heading Algorithm increases the weight of every detected heading card to ensure that the internal representation of heading cards reflects their significance within a user's spatial card arrangement.

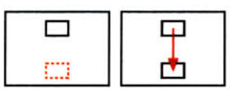
1.2.10. Addition /s



H: Every human thought impacts the creation of new knowledge and is stimulated either by previous thoughts or by novel experiences such as the input from the five senses and other nerve signals. While every thought alters the configuration of knowledge, the configuration of knowledge also affects future thoughts. In other words, while the current state of a human's knowledge structure determines where and how new information is added, the addition of new information in turn also modifies a mind's knowledge structure. Consequently, if the same piece of information were to be acquired twice, the consequent additions would be different in both cases. For example, if person A were informed by person B that tomorrows weather is going to be good, then person A might simply append this information to what he already knows. If later person A were given the same information by person C, then person A would not append this information a second time but more likely construct knowledge about the equivalent opinions of person B and C as well as the increasing likelihood about tomorrows weather being good. Hence, providing a person twice with the same information under the same circumstances does not produce redundant knowledge. Donald Schon's account of the reflective practitioner [27] considers the fact that nothing is ever experienced or seen in the same way as an important advantage for designers allowing them to easily reframe a problem or to create variations around the same theme. Since thinking itself produces new and unique information, the human mind also evolves in absence of external input by continuously reconfiguring existing knowledge. It is feasible to assume that in absence of external input, humans are focused on rethinking and refining what they already know and most likely end up repeating similar sequences of thoughts over and over again. On the other hand, in presence of external input humans are exposed to a world of data and information that may complement past knowledge in ways that lead to a more diverse range of views, ideas and procedures. An interesting hypothesis of what might happen to a person with minimal external input is outlined in a novel by Stephan Zweig [28] [29]. He portrays a prisoner in total darkness and isolation who tries to stay sane by memorizing and mentally replaying games of chess. In contrary to the scenario outlined in the novel, some people seek remote places with minimal exposure to external input that allow for concentrated thinking and self-reflection. While short periods of limited external input may be beneficial to the cultivation of existing knowledge, longer periods may lead to insanity or an unhealthy amplification of opinions and believes. In conclusion, the human knowledge structure evolves with and without external input. While the absence of external input fosters the reconfiguration of existing knowledge structures, the presence of external input triggers the integration of new data into existing knowledge structures.

C: The Addition Algorithm creates a directional relation between the most recently added card and the previously added, moved, modified or deleted card. In other words, the Addition Algorithm is modeled on the assumption that the current focus of attention is triggered by (or at least in some way related to) the previous focus of attention. Consequently, every new card leads to the addition of one permanent (static) relation. Every relation created by the Addition Algorithm is assigned an initial link weight of 5. Every consequent relation reduces the link weights of all previously added relations by 0.5. Because cards accumulate relations with other cards over time, newer cards often have fewer relations and as a result remain less noticeable for some time. The initially strong but fast diminishing link weight assigned by the Addition Algorithm ensures that new cards become instantly (though only temporarily) visible within the existing network of heavily interrelated cards and effectively reflect the user's current focus of attention and short term memory.

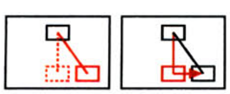
1.2.11. Deletion /s



H: The human brain does not have the capability to specifically "delete" pieces of knowledge. However, knowledge can be rendered obsolete through the creation of new and contradictory knowledge. The process of rendering knowledge obsolete is known as "forgetting" and considered a substitute for the computational term "deleting". In general, obsolete knowledge tends to be accessed and interconnected less frequently and subsequently looses its influence within the growing knowledge structure. Consider the following example: Person A trusts person B until person A notices person B is lying. The knowledge about person B being trustworthy is now in contradiction with the knowledge about person B being dishonest. The time it takes to outweigh person A's initial assumption about person B's honesty depends on many factors such as person A's devotion to this issue, the current strength of the obsolete knowledge within person A's existing knowledge structure, or person A's likelihood to consider people of person B's character as trustworthy. For example, person A might more intensively focus on this issue if heavily disappointed or personally affected by person B's dishonesty. If on the other hand person B were a very old friend of person A then the initial assumption about person B's honesty might be strong enough to supersede the more recent and contradictory knowledge addition. In this case person A may simply assume person B's behavior to be an exception. The time it takes to outweigh person A's initial assumption about person B's honesty also depends on the current rate and amount of new additions to person A's knowledge structure. This means that it is more difficult to forget something while there is nothing else to be concerned about. An attempt to actively focus on forgetting a particular piece of knowledge will most likely have an opposing effect since knowledge under the current focus of attention is expected to gain strength. Consequently, one would best forget about a particular piece of knowledge by intensively focusing on an unrelated problem or by constructing contradictory arguments. However, the more common and natural way of forgetting is by simply waiting until the unwanted knowledge looses its influence to more often used or to more recently acquired knowledge. Since in some cases unwanted knowledge is very distracting and painful, this long-lasting way of forgetting requires people to find ways to temporarily cope with unwanted knowledge (for example by trying to understand or by learning to accept unwanted knowledge). Since not only unwanted but all unused knowledge will weaken over time, the process of forgetting can eliminate essential and valuable knowledge as well. However, since forgotten knowledge is only weakened and consequently continues to exist in the highly complex and constantly growing knowledge structure, it is possible to revive (remember) forgotten knowledge even though often only with great difficulties. Note that the job of Psychiatrist is commonly one of helping people to remember and analyze past knowledge to understand the configuration of more recent knowledge structures.

C: The Deletion Algorithm executes three tasks: 1. The Deletion Algorithm labels deleted cards as "deleted". Deleted cards are invisible to the user but continue to exist in the database. Even though deleted cards remain in the database they are less likely to gain strength and consequently loose influence within the growing network of relations (analogous to forgetting). The advantage of preserving deleted cards is that the database history can be restored, that the relational structure does not suffer from missing nodes, and that deleted cards can be rediscovered. 2. The Deletion Algorithm creates a directional relation between the most recently deleted card and the previously added, moved, modified or deleted card. The established relation between the two most recently active cards represents and preserves the user's shifting focus of attention. 3. The Deletion Algorithm supplements every existing relation connected to a deleted card with an additional relation of negative link weight. This is to ensure that deleted cards instantly loose their influence over active cards. The initially assigned link weight of -2 is only temporary and reduced in steps of 0.2 over the subsequent 10 database modifications. This is to bridge the time until deleted cards loose their influence naturally due to their inability to accumulate new relations.

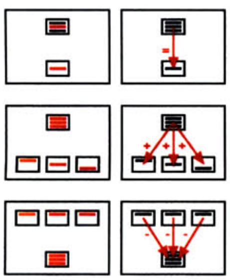
1.2.12. Modification /s



H: The analysis and operation of a physical, virtual or even mental information space not only requires a good overview and understanding about the relations among information objects but also about the information space history such as the date and time (sequence) of additions, deletions and modifications as well as authorship and other contextual information associated with individual information objects and relations. In other words, to fully understand an information space, one has to recognize as well as to keep track of changes to the information space. Additions and deletions are usually more easily recognizable than modifications. Consider a file system maintained by multiple administrators. An individual administrator may be more likely to notice the addition or removal of an entire file than a minor contextual modification to a file. Modifications are usually only noticeable if clearly indicated (e.g. red colored hand notes in a typewritten document). For all administrators to stay current on modifications to files in a file system requires frequent communication among the administrators or a carefully maintained log listing every change to the file system.

C: The Modification Algorithm deals with card modifications on a user's Work Area. Card modifications include changes to the content as well as changes to the size and location of cards. The Modification Algorithm maintains a copy of both the previous and the current version of a modified card. The Modification Algorithm applies the functionality of the Deletion Algorithm to the previous version of a modified card to preserve the card history. The Modification Algorithm then applies the functionality of the Addition Algorithm to the current version of a card to ensure that modifications and additions are treated equally within the relational network. Finally, the Modification Algorithm adds a relation with a link weight of 2 between the previous and current version of a modified card to establish a strong and permanent connection between the two card versions. This algorithm is essential for the preservation of the card arrangement and authorship history.

1.2.13. Replication /s



H: Humans commonly create new knowledge and information based on what they already know or based on information and data that already exists. For example, a student composing a paper on Roman history is most likely to read, combine and reformulate fragments from previous papers on Roman history. Optimally the student would reference all information sources so as to give credit to the authors as well as to point the reader to more detailed information about the subject. Similarly, computer programmers assemble code fragments from software libraries when developing new applications. Also architects and designers often combine and modify existing design ideas as a basis for the creation of new and unique design solutions. The combination and recombination of information has become a standard practice due to the waste availability and accessibility of information. Until recently a reference would only indicate the source but not the content of information. However, with the introduction of the hyper-text-markup-language (.html) [30] readers can gain instant access to the content of referenced web-based resources. The more recently introduced extensible-markup-language (.xml) [31] allows authors not only to reference but also to combine content from various web-based information sources. Documents composed from .xml content dynamically reflect all changes to the original information sources.
The DataBox project [32] (a predecessor of the EWall project) proposes an inclusive computational solution for the reuse of web-based content. The DataBox Inserter allows authors to compose web-based documents by directly inserting content from web-based documents created by other authors. Small icons indicate and hyper-link the location of remotely stored content. Source documents can themselves contain remotely stored content consequently allowing readers to recursively explore the compositional history of documents. The DataBox Explorer searches documents for the presence of specified key words and complements the search results with detailed information about directly and indirectly related documents (such as parent and child documents). The research concluded significant advantages for tracing the heritage of information as well as opportunities for preserving authorship rights.

C: The Replication Algorithm keeps track of and relates cards that contain content from other cards or that are duplicates of other cards. If a user copies content from one or more cards into one or more cards or if a user duplicates a card then the Replication Algorithm adds relations between the source and destination card according to the following three rules: Rule 1: A Substitution occurs if a user cuts or copies all or parts of the content from one card into another card or if the user duplicates a card. A Substitution concludes with a directional relation from the source to the destination card with a (neutral) link weight of 0. Rule 2: A Variation occurs if a user cuts or copies all or parts of the content from one card into two or more other cards. A Variation concludes with one directional relation from the source card to every destination card with a (positive) link weight of 1. Rule 3: A Combination occurs if a user cuts or copies all or parts of the content from two or more cards into another card. A Combination concludes with one directional link from every source card to the destination card with a (negative) weight value of -1. The reasoning for the relations and link weights is the same as for the Process Algorithm (see 1.1.4.). Relations created by the Replication Algorithm are merged with relations created by other algorithms but can also be separated to analyze the heritage of information and authorship history.

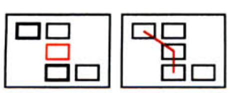
1.3. Level III Algorithms

Level III Algorithms create relations based on the collaborative use of information such as the exchange of information between multiple users or the access of information in a shared database.

H: The human ability to creatively interpret incoming signals is essential for the conversion of data into information and information into knowledge. The interpretation of data and information varies depending on the interpreter, the content, and the contexts in which data and information is created, interpreted, and used. As data and information are subject to continuous change and because different people usually interpret data and information differently at different times and under different circumstances, the results of such interpretations are often unpredictable and incompatible. The many possible interpretations of data and information is often referred to as "ambiguity" or "polysemy" and commonly considered a disadvantage for the accurate interpretation of data and information. It is due to this ambiguity however that humans develop new ideas and find novel solutions to problems. It is also due to this ambiguity that the collaboration between people often produces faster and more effective solutions resulting from the multiple perspectives on common tasks. Unlike humans, computers follow strict protocols when dealing with ambiguous cases during the interpretation of data and information and consequently are less likely to produce unexpected and unique results. Consider for example a database query for a book title. A computer would return results that match a particular set of key words. A human however can more feely explore the bookshelves of a library and return with a set of books that are relevant yet do not necessarily match the initial search criteria's. The interpretation of data and information also depends on validity, relevance and priority. For example, a statement such as "Fire-trucks are red" may not be true. To determine the validity of such a statement one might either evaluate the reliability of the person making the claim, confirm with a trusted person or the opinions of a large number of people, investigate the color of fire-cars on a large enough number of samples, or test if the claim corresponds with previously acquired knowledge. The relevance of data and information is based on the interpreters' focus and background. Consider for example a blue print for a building. The architect might be very concerned about the artistic aspects of the design while the engineer might focus on the structural components. Even though the architect and the engineer share the same information their focus and consequent interpretations differ. The priority of data and information is often unrelated to its validity and relevance. For example, a sign indicating "Danger" is likely to receive priority over notifications that are of more immediate interest to the reader. Finally, the interpretation of data and information is also influenced by who uses or used data or information where, when and why. Understanding the evolution of data and information is invaluable for envisioning and validating potential interpretations. For example, it may be helpful to know that a particular piece of information has previously been used for the resolution of a problem similar to the one at hand. In conclusion, the interpretation of data and information depends on a wide variety of factors that rarely ever produces predictable and compatible results. In other words, every interpretation is a customized view of a particular piece of data or information that might only be meaningful under specific circumstances and to an individual or a particular group of people.

C: Level III Algorithms discover potential relations between cards based on the analysis of the direct exchange of cards between users as well as based on the indirect exchange of cards through the use of shared databases such as database additions, modifications and retrievals. Level III Algorithms keep track of and analyze who exchanges which cards where, when, why, and with whom. Based on this analysis Level III Algorithms attempt to build associations among users, cards and contextual circumstances. The relations constructed by Level III Algorithms provide a basis for the operations of various Transformation Algorithms. Examples may include determining the validity, relevance and priority of cards from the viewpoint of individual users or groups of users as well as determining the foci, interests and background of individual users.

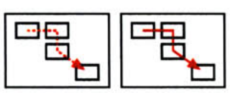
1.3.1. Focus /s



H: Humans often associate knowledge intuitively. For example, knowledge associations can be created based on the simultaneous input from the five senses. Imagine a person accidentally hurting himself while listening to a particular piece of music. In the future, this piece of music is most likely to remind this person about his accident. Hence, this person intuitively created an association between the simultaneous inputs from his touch and hearing senses. Intuitive knowledge associations can also be created between sensory inputs and thought processes. For example, a person might associate the content of a discussion with a particular smell that was present at that time. Finally, intuitive knowledge associations can also be created between multiple parallel thought processes. For example, a person talking on the phone while simultaneously working on a computer might create associations between the content of the phone conversation and his current work task. The relations established through the interaction among multiple stimuli and thought processes are in many ways similar to the consensus built through the interaction among multiple collaborators. In other words, individual (intra-personal) and social (inter-individual) thought processes bear many communalities. The thoughts of individual collaborators are continuously exposed to and consequently influenced by the contributions of other collaborators. Therefore, the individual contributions from various collaborators within a certain time period are often related. Even though individual collaborators have different foci, interests and backgrounds, their understanding of a common task as well as their colleagues' way of thinking will likely advance and equalize over time. Pierre Levy and Robert Bononno [33] refer to this phenomenon as a collective intelligence that will advance through information technology and eventually allow for an unfettered exchange of ideas in cyberspace thus liberating humans from social and political hierarchies. In conclusion, the convergence of input from various different sources exceeds the sum of its parts meaning that the independent assimilation of data and information as well as the independent development of thought is missing out on potentially relevant and interesting relations between inputs from different sources. In other words, parallel input from different sources fosters opportunities for new and creative relations to emerge.

C: The Focus Algorithm maintains a dynamic list of focal points inside a shared database of cards. A focal point is a reference to a card that has recently been accessed by a user or an algorithm. For example, a focal point might reference a specific card because it was recently created, modified or examined, because it emerged from a recent search query, or because it was recently related to another card. The Focus Algorithm also continuously creates relations between cards that are referenced by focal points. These relations highlight potential commonalities among cards that have been the focus of attention by different users or processes over the same time period. Relations created by the Focus Algorithm reflect on and preserve some of the database context (state of mind).

1.3.2. Footprint /s

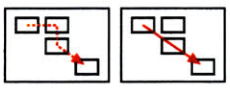


H: The accessibility of individual human memories varies. For example, recently acquired memories as well as frequently accessed memories are typically easier to recall. Humans also seem to follow specific pathways when exploring their own memories. For example, a pleasant memory about an evening on a fireplace may well trigger an unpleasant memory about an accident caused by a fire. From these two examples we may assume that human memories are not randomly accumulated but meaningfully structured and networked according to relations of varying significance (strength). The following analogy offers a possible model of human memory organization that provides a feasible explanation for the previous two observations: Imagine the human memory as a road map with intersections representing memories and roads representing relations between memories. The most recently recalled or created memory represents the current location on the road map. A journey through this network of memories provides the traveler with a particular piece of information at every intersection. At every intersection the traveler has to decide on which road to proceed. A road between two intersections may be a heavily accessed highway, a frequently accessed street or an occasionally accessed path. A typical traveler might be more likely to choose the biggest road and less likely to choose the one he just came from. A traveler might also choose a small path or randomly pick a road hoping for interesting discoveries. A traveler may even back off one or more roads if the direction turns out to lead away from his current focus or interest. The size of a road (or number of footprints) increases with every access making the road even more attractive to travelers during future explorations. The road map metaphor highlights interesting similarities with human memory access: 1. Recently acquired or discovered memories are more likely to be accessed because these intersections are close to the current position of the traveler. 2. Frequently accessed memories are more likely to be accessed because these intersections are connected either through many or through big (and consequently more easily accessible) roads. 3. The same pathways through memories are often accessed repeatedly because travelers have a tendency to stay on big roads. The road map metaphor not only shows similarities with human memory but also social and organizational networks as well as physical and virtual data structures. Most noticeable is the strong tendency of humans to follow their own footprints as well as the footprints of other people. For example, a human is more likely to show interest in, and less likely to question the content of a new book if it has received a good evaluation from a large number of people, if it is referenced in various other books, if it is written by a popular author or if it is displayed in a prime location in a book store. Such a book will sell more frequently and in turn become even more popular. Within the road map metaphor such a book would be located at the intersection of various heavily accessed highways. Such an intersection would encourage a lot of traffic and in turn also increase the number and size of roads within its vicinity. It is reasonable to assume that once an intersection has gained some significance it would continue to increase in popularity until its strength outbids all other intersections (similar to big corporations whose monopolies eliminate competition). However, the popularity of an intersection not only depends on the number and size of connecting roads but also on the number and distance of potential visitors. This means that if no travelers' are located within the vicinity of a heavily connected intersection the traffic remains low. In other words, if the focus and interest of travelers changes then even the most elaborate infrastructure may loose out on visitors and eventually give in to competing infrastructures. For example, a prime location for a store in a city is of little value on rainy days when most habitants stay home. Also, a popular book may no longer be read because it might have lost its relevance due to changing circumstances (context), because some of the more explorative readers might have discovered new books that eventually would become more popular, or because the main ideas are adapted and elaborated in other books. In conclusion, the footprint algorithm increases the link weights between nodes in dynamic networks similar to the accumulating footprints created through the movement of people.

C: The Footprint algorithm enhances frequently accessed relations between cards during database searches, random explorations as well as additions and modifications. A Footprint relation is added between two cards if the same person or algorithm accessed the two cards sequentially. These additional relations will increase the link strength between two cards making it more influential during future database accesses. Footprint relations basically visualize database traffic and dynamically highlight popular cards and database sectors. The automatic addition of such relations is especially beneficial for the analysis and access of collaborative databases.

Note: The web search engine Google [34] and the retail store Amazon [35] operate based on similar concepts. Google establishes relations among web sites that have been accessed by the same users consequently being able to provide users not only with search results of web sites that match a specific query but also with a selection of web sites that other users accessed while exploring similar issues ("Similar pages"). Amazon establishes relations among products that have been purchased by the same customers thus being able to provide customers with a list of products related to the products they previously bought or are currently considering buying ("Customers who bought this [product] also bought ...").

1.3.3. Query /s



H: When humans "think about something" their conclusions are associated with whatever motivated this mental exploration, with details on how they arrived at their conclusions as well as with information on the success of the conclusions. The initial objective and concluding discoveries become consequently related and may well direct future explorations. For example, a human might investigate whether a statement he just read in a research paper might support some of his previously acquired knowledge. The consequent mental exploration is likely to pass through a large number of mental pathways evoking many knowledge items along the way. Optimally, the investigation would terminate once a set of knowledge items are discovered that can be used to either confirm or contradict the statement from the research paper. The newly acquired knowledge from the research paper will then have established relations with the conclusions from the metal exploration as well as the experiences made during this investigation. Any future investigation touching on the knowledge retrieved from the research paper is likely to reactive the conclusions from the previous investigation (and vice versa).

C: The Query Algorithm constructs relations between starting points, waypoints and endpoints of a database queries or explorations. The starting point refers to the most recently accessed card that inspired an investigation. The endpoint refers to a card that terminates an investigation. The waypoints represent all the cards investigated while progressing from the start to the end point. The link weights of all relations established between the individual waypoints and the endpoint gradually increase from 0 (at the starting point) to 1 (at the endpoint). The gradual increase of link weights indicates the increasing relevance of cards while closing in on a probable solution. The relations established by the Query Algorithm will most likely influence future database explorations by increasing the accessibility of these end points. Because the Query Algorithm has no means of differentiating between endpoints that represent satisfactory and unsatisfactory solutions, the increase in accessibility to such endpoints may be misleading in some cases. However, the early discovery of unsatisfactory endpoints may also prevent the repetition of previous mistakes and limit the search space to more promising areas of investigation.

1.4. Level IV Algorithms

Level IV Algorithms are designed to emulate how humans, consciously or unconsciously, free associate when thinking by themselves, without external input.

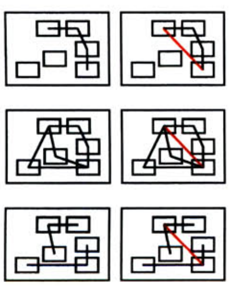
H: Humans not only accumulate and associate knowledge by interpreting incoming signals but they also reflect on what they do, feel and know without external input, through thinking. Thinking can happen consciously or unconsciously. Conscious thinking takes place with a goal in mind such as when trying to understand the meaning of life. Unconscious thinking happens for example while dreaming (see Note below). Thinking in absence of external input is often less effective. Consequently, many people prefer to discuss their thoughts with others to acquire alternative views and feedback. However, the human ability to mentally elaborate (to think something over) in absence of external input is often useful to organize, reflect and understand previously acquired knowledge. Furthermore, frequent isolated thinking may prove essential to maintain a healthy balance between accumulating new knowledge and structuring existing knowledge. The evolving human mind dynamically changes the way humans perceive, understand and respond to their environment. Due to this continuously changing knowledge structure, humans are unlikely to see, comprehend or do anything twice precisely in the same way. While this may sound somewhat limiting, it is the basis for human creativity and adaptability allowing humans to develop solutions to new problems and to cope with unfamiliar situations. Furthermore, the dynamic self-reorganization of the mind helps to prevent humans from getting stuck with a particular problem. While a computer would repeatedly analyze a problem in the same way, humans continuously evolve during a problem solving process allowing for new perspectives and possible solutions to emerge. This is why the solution to a problem may seem inaccessible at one point and suddenly materialize at a later time. Consequently, to distance oneself temporarily from an unresolved problem is often more effective than to insist on an immediate resolution. This theory goes in hand with expressions such as "sleep it over", "keep it in the back of your mind", "shake up", "get unstuck", or "find a fresh view". Each of these expressions indicates a bottleneck during a problem solving process and suggests either to review the problem at a later time or to look at the problem from a different perspective. While postponing the problem resolution may seem convenient it does not guarantee success and only is an option if time is not an issue. Looking at a problem from different perspectives may require an experienced person with the ability to temporarily disregard the conclusions from previous investigations or the varying views of multiple people.

Note: Dreaming: Over the past century, research on dreaming has produced a variety of interesting yet inconclusive theories. For example, in 1900, Sigmund Freud published a paper indicating that dreams reveal unconscious worries and desires [36]. In the 1960's, William Dement found that dreams result from a sleeping person's inability to respond to messages sent from the brain stem to the visual center of the cortex [37]. "In 1977, Allan Hobson and Robert McCarley proposed that dreaming was the brain's attempt to respond to stimuli [38]." [39] In 1983, Francics Crick and Graeme Mitchison argued "that dreaming was the process of discarding unwanted memories. They reasoned that the signals sent to the cortex wiped out unneeded information

[40]." [39] In 1985, "Jonathan Winson suggested that dreaming allows the brain to process daily experiences and apply them in "an ongoing strategy for behavior" [41]." [39] Also of significance is Gaston Bachelard's [42] distinction between the unconscious activity of dreaming versus the conscious activity of day-dreaming (reverie).

C: Level IV Algorithms construct relations based on the analysis of previously established relations and in complete isolation of any input source. These relations are conceived during system idle times allowing the database-structure to advance without external input, instructions or feedback. Level IV Algorithms primarily conduct simple cleanup operations without a higher-level purpose. These cleanup operations during system idle times can be compared with the unconscious thought processes of humans such as the theory of neural garbage collection during sleep. Level IV Algorithms are intended to focus on tasks that are neglected during normal (focused and time-constrained) operations such as the exploration of weak and rarely accessed link structures, the evaluation of alternative routs between indirectly connected nodes, or the discovery of similarities among nodes and network patterns. Level IV Algorithms may be envisioned as mechanisms that gently shake the database content once in a while to change the balance of relations in a meaningful way. Even though Level IV Algorithms do not modify but only complement existing relations, equivalent database searches before and after Level IV operations are likely to produce different results. Level IV algorithms are supposed to ensure a flexible and dynamic data structure that prevents the long-term domination of strong network segments (local maxima) and allow for more explorative and inspiring data discoveries.

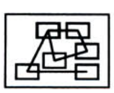
1.4.1. Shortcut /d

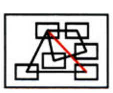


H: Another quality of the human mind is its ability to make assumptions (inferences). Assumptions allow humans to deal with situations in which not enough information or knowledge is available to accurately determine the correct answer to a question or the best course of action. Because assumptions can be wrong they are more valuable in situations with not too much at stake or with few alternatives to choose from. Assumptions are constructed based on past experiences involving concepts such as similarity, frequency, or indirect relationships. Consider the following examples: Similarity: e.g. to notice that the volume of water increases corresponding to its weight may lead to the assumption that the volumes of other substances also increase corresponding to their weights. Frequency: e.g. to repeatedly notice a high traffic volume in the mornings may lead to the assumption that the traffic volume is always high in the mornings. Indirect relationships: e.g. to know that somebody was hurt during an encounter with a snake may lead to the assumption that the snake was the cause of this incident. In conclusion, assumptions constitute emerging and unique concepts that are constructed based on existing knowledge, that are triggered by current circumstances and that represent potentially feasible answers to questions or possible courses of action.

C: The Shortcut Algorithm analyzes the existing structure of relations for the possible addition of shortcuts. A shortcut can be a relation between two indirectly linked nodes (1.4.1.1), between two nodes connected through multiple pathways (1.4.1.2), or between two heavily linked nodes (1.4.1.3). Consider the following possibilities: 1. If several relations exist between node A and B, and if several relations exist between node B and C, and if no relations exist between node A and C then the Shortcut Algorithm would construct a relation between node A and C assuming that if there is a strong communality between node A and B as well as node B and C then it is likely that there is some communality between node A and C. 2. If no relation exists between node A and E but several pathways that indirectly connect node A and E (e.g. A-B-C-D-E, A-F-E, and A-G-H-I-J-E) then the Shortcut Algorithm would construct a direct relation between node A and E. 3. If there is not relation between any two heavily liked nodes then the Shortcut Algorithms would construct a relation between these two nodes assuming that two major access points (knowledge domains) ought to be connected. The Shortcut Algorithms can be compared with a road construction authority that evaluates new connections between villages, towns, and cities based on the current layout and access frequencies of roads, streets and highways. Consider the following examples: 1. The road construction authority might decide to build a road between village A and C if the roads between village A and B as well as village B and C are heavily used. In this case the road construction authority might assume that a lot of traffic between village A and B as well as village B and C are actually caused by people traveling between village A and C connecting through village B. 2. The road construction authority might decide to build a street between town A and E if several roads and villages located between the two towns experience a high traffic volume. In this case the road construction authority might assume that many people traveling between town A and E currently have to find their way through several roads and villages located between the two towns. 3. The road construction authority could decide to build a highway between two cities. Cities commonly provide access to a large number of towns and villages. Consequently, the highway would not only serve people traveling between the two cities but also between the towns and villages located near the two cities.

1.4.2. Discovery /d

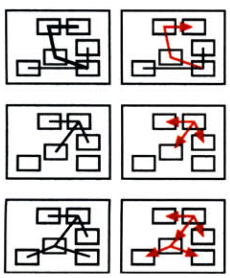




H: Some human knowledge seems to be recalled more easily than others. This is because certain knowledge items are more heavily interconnected with other knowledge items, and consequently accessed more frequently. To access less interconnected knowledge items one would have to concentrate on exploring less obvious mental pathways, a process often referred to as "remembering". Once a particular knowledge item is remembered it might become interconnected with more recent or related knowledge items thus increasing its accessibility within the knowledge structure. Less interconnected knowledge items are also explored during a "brainstorming" activity, a random exploration and evaluation of knowledge items that may support the resolution of a particular task or problem. In conclusion, the frequent exploration of less interconnected knowledge items can be useful to reevaluate and reactivate knowledge items that might not have been of value or not have made sense at the time of their acquisition.

C: The Discovery Algorithm randomly explores the network structure for weakly related nodes (a weakly related node is a node whose combined weights of all attached relations is very low). These nodes are being related with more recent additions so as to stimulate their discovery. Relations established by the Discovery Algorithm loose their strength over time and eventually disappear. In other words, the Discovery Algorithm only increases the accessibility of weakly related nodes on an occasional and temporary basis. During this time period weakly related nodes are more likely accessed by Interpretation and Transformation Algorithms thus increasing their chances of regaining strength within the network structure and of being discovered during database searches. The Discovery Algorithm can be seen as a mechanism that moves the focus of attention during times of inactivity (the focus of attention usually points to the most recently active node). The Discovery Algorithm recursively explores nodes that are directly related to its present location. The Discovery Algorithm first examines nodes connected by weak relations and only backs off (terminates the recursion) if a previously examined node or a terminal node (a node related to only one other node) is encountered.

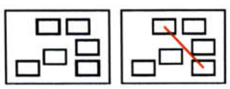
1.4.3. Concept /d



H: The human mind can be seen as a complex network of knowledge items and associations. While every individual association indicates a relationship between two knowledge items, certain combinations of associations can indicate the possible presence of a concept such as a process or a grouping. In this context, a process refers to a sequence of events such as the order of activities required to startup a car. A grouping means a set of belonging components such as the ingredients for a cooking recipe. The ingredients have little in common but the fact that they belong to the same recipe. Research on how the human mind memorizes, recalls and compares such concepts is broad and speculative. For example, Marvin Minsky [3] proposes that the combination of all active memory cells at a particular moment in time could represent a concept. Such a concept would be recalled whenever a similar combination of memory cells is activated. This theory suggests a higher-level function that can memorize, recall and compare the various combinations of active memory cells. If the mind had no higher-level functionality but could only deal with knowledge items and associations then the memorization, recollection and comparison of concepts would seem even more complex. Associations among knowledge items that belong to the same concept would be indistinguishable from other types of associations. Only minor variations among the patterns of associations in different parts of the network might hint the possible presence of a concept.

C: The Concept Algorithm searches for and amplifies network patterns that might represent a concept. Consider the following possibilities: A Sequence (1.4.3.1) is assumed if a series of nodes is connected with relations of equally or similar types and weights. For example, three relations of type 1 and weight 4 connect node A and B, B and C, as well as C and D. A, B, C and D might be related with several other nodes, though not through relations of type 1 and weight 4. The Concept Algorithm complements the relations of a sequence with one directional relation between each node pair thus increasing the "visibility" and "life-expectancy" of the sequence. All directional relations point away from the starting point of the sequence. The Concept Algorithm assumes that the starting point of a sequence is referred to from within several network domains thus making it the most heavily related node within the sequence. A Group (1.4.3.2) is assumed if one node (the parent-node) is connected to multiple other nodes (the child-nodes) with relations of equal or similar types and weights. The Concept Algorithm complements the relations of a group with directional relations that point towards the child-nodes. A Hierarchy (1.4.3.3) is a combination of multiple groups meaning that a child-node of one group can be the parent-node of another group. The Concept Algorithm complements the relations of a hierarchy with directional relations that point towards the child-nodes though decreases the weights of the directional relations with increasing distance from the top-most parent-node. The presence of a hierarchy might indicate a hierarchical order of knowledge items or a goal-oriented process. For example, a hierarchical order of knowledge items might separate vehicles into cars and trucks, and trucks into fire trucks and pickup trucks. A goal-oriented process might hierarchically outline the development of a problem-solving activity. For example, step A lead to step B and step B lead to step C. Step C failed thus causing the return to step B and the consequent development of step C1.

1.4.4. Similarity /d



H: Humans are capable of comparing new knowledge with existing knowledge, thus allowing for the creation of meaningful associations among related knowledge items. Imagine a person reading several books. The person continuously compares the content of every newly read book with the contents of previously read books. This person might detect interesting similarities and contradictions among the different books, consequently constructing a knowledge network that may be more dense and interconnected than without comparing the books. For example, the person might read in one book about a shipwrecked individual being stuck on a tropical island. Later this person reads in another book about a romantic tropical island vacation. The comparison of both stories will allow the person to associate tropical islands with both, its dangers and beauties. (Clifford Geertz's [43] work on "thick descriptions" provides a more detailed investigation into the issue of associating information with multiple different meanings.) In conclusion, to carefully compare new knowledge with existing knowledge can allow for the creation of meaningful associations, can prevent the storage of redundant knowledge within different knowledge domains, and can encourage the continuous evaluation of the validity and feasibility of new knowledge.

C: The Similarity Algorithm relates cards that contain similar text in their icon or heading areas, or that point to the same URL. While a comparison of all words in a text (including articles like "the" and verbs like "can") is unlikely to produce reasonable results, this algorithms only considers nouns, capitalized words, and words that do not appear in a standard dictionary. The Similarity Algorithms assigns relations of differing weights to reflect the level of similarity. The relation weight increases by 1 for every pair of nouns, by 2 for every pair of capitalized words, and by 3 for every pair of words that do not appear in a standard dictionary. The effectiveness of this algorithm can easily be improved by drawing from commercial applications and advanced research on recognizing text similarities such as the work by Ozlem Uzuner et al. [44].

2. Transformation Algorithms

Transformation Algorithms leverage the relations constructed by the Interpretation Algorithms to help users monitor, search, exchange, and visualize information. Transformation Algorithms are separated into four distinct groups: News, Database, Exchange, and Visualization Algorithms. News Algorithms help users to monitor for additions and changes to selected information sources. Database Algorithms help users to search databases for relevant information. Exchange Algorithms help users to prioritize information exchanged among users. Visualization Algorithms help users to visualize information. Every Transformation Algorithm consists of a Recognition Function and an Execution Function. Recognition Functions retrieve and analyze relations among information items that are compatible with particular Execution Functions. Execution Functions select, prioritize, and organize information for individual users.

H: Transformation Algorithms are designed based on how humans use knowledge to act and how they assess such actions. More specifically, the design is based on how humans bring knowledge to consciousness and make knowledge selections that help them deal with current needs and circumstances. The retrieval of human knowledge is primarily triggered by sensory input (experiences). In other words, what humans see, smell, hear, taste, and feel triggers the activation of particular knowledge. Due to differing pre-associations, different humans revoke different knowledge under similar circumstances. Furthermore, due to the continuously changing backgrounds of individuals, humans are unlikely to revoke the same knowledge under repeating circumstances. Marvin Minsky [45] elaborates on this issue by indicating that "The secret of what something means lies in how it connects to other things we know. That's why it's almost always wrong to seek the real meaning of anything. A thing with just one meaning has scarcely any meaning at all." This statement not only supports the notion that human experiences are interpreted differently by different individuals but also that concrete and unambiguous interpretations of experiences are neither feasible nor desirable. Consider the following two extremes: If human experiences could be accurately shared and preserved then humans would not be able to engage in conversations or expose themselves to new views and ideas. If on the other hand the continuously changing interpretations of experiences would differ fundamentally then humans would have no reason to share their opinions or conclude their thought processes. Thus, the ability of the human mind to allow for interpretations of experiences that are neither too concrete nor too ambiguous is a fundamental asset that allows humans to coexist and advance. Concrete interpretations of experiences are more likely to trigger a top-down approach of thinking (removing constraints) meaning that concrete experiences comply with existing knowledge and are subsequently assumed to have a higher likeliness of being valid or useful. In other words, concrete experiences are perceived as a basis upon which (if time allows) options can be explored. For example, a card driver would initially proceed as indicated by the traffic signs before examining the situation and consider other options. Similarly, pilots follow strict procedures (check-lists) and only apply individual alterations in unforeseen situations. Ambiguous interpretations of experiences are more likely to trigger a bottom-up approach of thinking (adding constraints) meaning that humans evaluate and combine existing and emerging components from large numbers of possible interpretations of experiences. For example, designers often start out with a wide range of possible design ideas that eventually combine into one final solution. This final solution is usually not a recombination of previous solutions but something new and unique. Thus, ambiguous interpretations are essential for creative and innovative tasks such as designing, brainstorming, problem solving, and decision-making. The human ability to develop ambiguous interpretations from experiences usually decreases with age. The reason for this is the continuously increasing knowledge structure of the human mind. With age, humans increasingly preference concrete experiences that do not conflict with what they already know and that do not require them to reconsider what they already know. While this development may cause a loss of flexibility and creativity it is also what makes humans unique and what is commonly referred to as character, style, individuality, and personality. Theoretically, if humans were to live forever, they would eventually arrive at a point where they would end up with only one opinion for every issue and become incapable of adapting and innovating new views and ideas. Thus, aging people that depend on their abilities to innovate and to remain mentally flexible must learn to develop the mental techniques to temporarily disengage themselves from many things they know in order to be able to develop and recognize new and unique views and ideas.

C: Transformation Algorithms compare the contents on the users' Workspace Views with the network of information established by the Interpretation Algorithms and subsequently collect, prioritize, and arrange information relevant for specific users. The propositions made by the Interpretation Algorithms may be viewed as (and compared with) propositions made by human contributors. Every individual Transformation Algorithm independently interprets the relevance of information from its unique perspective. Thus, different Transformation Algorithms may develop differing propositions about the relevance of information. The presence of many differing propositions indicates ambiguity while the presence of many complementary propositions indicates concreteness. Unlike Interpretation Algorithms, Transformation Algorithms do not detect implicit relations, meaning that Transformation Algorithms develop their propositions solely based on the analysis of relations established by Interpretation Algorithms. Users can choose from four options that define how the individual Transformation Algorithms negotiate their propositions: The first option is to preference propositions that are shared by the majority of algorithms. The second option is for users to manually adjust the level of influence for individual algorithms. The third option is for users to adjust the level of influence for individual algorithms through the use of a training program. The fourth option is for the individual algorithms to autonomously adjust their levels of influence based on direct and indirect user feedback [46].

2.1. News Algorithms

News Algorithms inform users of relevant additions and modifications to specified information sources. For example, if a user were to specify several web based news papers as his information sources, then the News Algorithms would compare every new addition and modification to these newspapers with the contents on the user's workspace. The user would be presented with a list of cards each representing one newspaper addition or modification of potential relevance.

Note: The use and operation of News Algorithms is explained in Section A (News Module).

Note: This work does not introduce the mechanics of the individual News Algorithms as most of it is based on conventional technologies.

H: News Algorithms are designed based on how humans monitor for and determine the relevance of new information. Whether and how humans respond to new information depends on a variety of factors such as how new information is presented and whether new information intersects with the interests and needs of the viewers. For example, an advertising poster might appeal to the viewers because of its layout, its content, or its associated brand name. There are a variety of ways to monitor for new information. For example, an easy way to stay current with new information of general interest is to read newspapers and magazines. A more customized exchange of new information happens through the interactions among office workers that approach their office mates with new information that they find interesting or that they consider of potential relevance to their peers. People in many professions heavily depend on access to new information. For example, stockbrokers have to continuously retrieve and quickly process large amount of new information to adjust their predictions of future events. In conclusion, the most essential activities in dealing with new information include how to gain access to new information, how to deal with large amounts of new information, how to determine the relevance of new information, and how to estimate the credibility of new information.

C: News Algorithms monitor for recent and potentially relevant additions and modifications to information sources such as databases and web sites. Unlike Database and Exchange Algorithms, News Algorithms do not determine the relevance of new information based on previously established relations (Interpretation Algorithms establish relations based on the usage history of information and subsequently are ineffective on new information) but based on similarities between the contents of new information and the contents of information users are currently working with. News Algorithms also estimate the potential credibility and user interest of new information based on whether new information comes from sources or compares with subjects that users frequently adopted new information from.

2.2. Database Algorithms

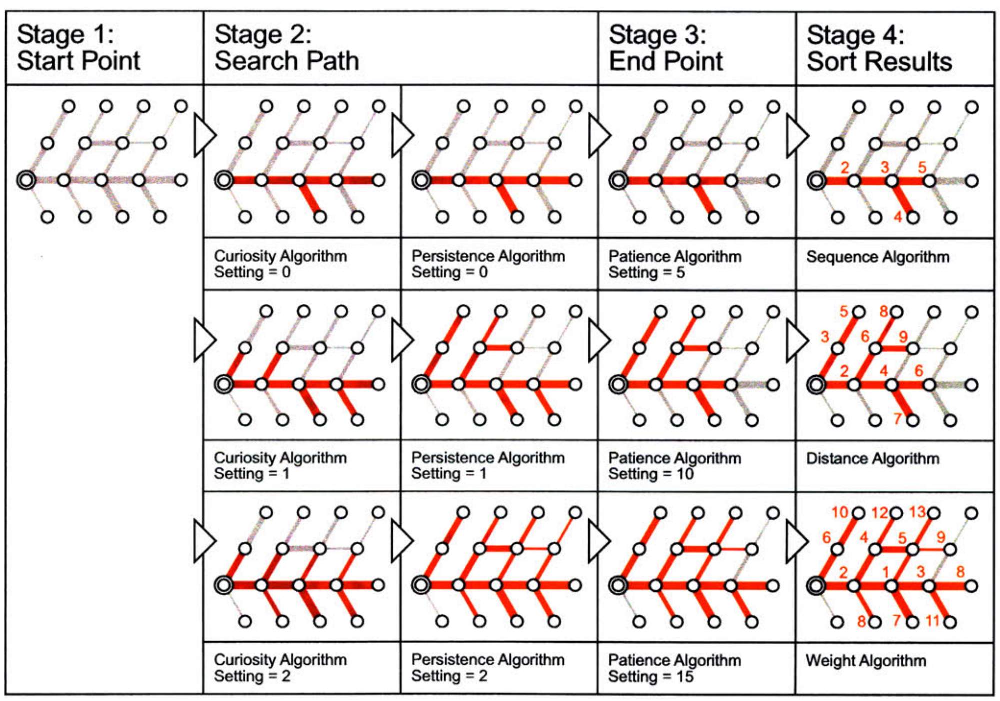
Database Algorithms retrieve user relevant information from databases whose contents are networked. The goal of the Database Algorithms is not to detect information that precisely matches but relates to the interests and foci of particular users. Conventional database searches commonly return information with contents corresponding to one or more keywords in a search query. One problem with this approach is that relevant information may not necessarily contain words that match a specific search query. For example, a search for VHS recorders is unlikely to return information on DVD burners even though DVD burners offer a feasible alternative. Database Algorithms retrieve user relevant information based on the relations among information items as a means to complement existing technologies that focus on information contents.

Note: The use and operation of Database Algorithms is explained in Section A (Database Module). Although there are many possible ways of exploring networked database structures this work only introduces a small set of Database Algorithms as a means to explain the concept and as a basis for further exploration.

H: Database Algorithms are designed based on theories of how humans bring knowledge to consciousness when recalling, comprehending, and generating information. How knowledge is accessed and explored depends on a particular human's knowledge structure, the methods of exploring the knowledge structure, and the circumstances that stimulate such explorations. Every exploration of knowledge starts at a particular place in the human mind, progresses along mental pathways, terminates under specific circumstances, and produces multiple results that need to be evaluated and prioritized. Research on how such processes work is inconclusive and speculative. Cognitive scientists commonly describe such processes within the context of mental networks that are composed of nodes and relations [47]. Consider the following small experiment: A person is asked to listen to a word and immediately respond with another word that comes to his mind. For example, the person might respond to the word "blue" with "color", to "forest" with "green", and to "green" with "color". These responses indicate previously established relations among the meanings of the words and also highlight the dominant position of the word "color" within the mental network. The configuration and navigation of mental networks are unique for every person and fundamental to how humans think and act.

Illustration 4

Database Algorithms Scenarios



C: Database Algorithms explore the contents of networked database structures for relevant information. Some of the designs draw from research on querying semi-structured data [48] [49]. Four different groups of Database Algorithms (/s /p /e /r) are applied in sequence to handle activities similar to the human knowledge retrieval process:

Stage 1: Define a Start Point (/s1)

The first group of Database Algorithms is responsible for specifying the Start Point of a database search. The Start Point references a node in a database from where directly and indirectly connected nodes are being explored and whose content is related or relevant to the work of a particular user. The Start Point may be determined through a conventional database query such as a text or an image [50] search. A Start Point might also reflect an active node in a database such as the most recently added, retrieved, modified, or traversed node. Furthermore, a Start Point might point to so-called "Stimuli" or "Focal Point" that hint areas of common interest and that are dynamically relocated by Interpretation Algorithms (compare Section A / Level III and IV Algorithms). Start Points dynamically change their locations and multiple Start Points may initiate simultaneous searches that originate from different locations (compare bi-directional search [51]).

Stage 2: Define a Search Path (/s2)

The second group of Database Algorithms defines the Search Path. The search path determines in what order the nodes in the database are explored. A search initiates at a Start Point and proceeds with the evaluation of directly connected nodes. By default, only the most heavily connected nodes are considered and previously encountered nodes are ignored. The Database Algorithms also offer options for the examination of less heavily and indirectly connected nodes. For example, the Curiosity Algorithm is used to increase the probability for the examination of weaker branches and the Persistence Algorithm for the examination of nodes within a wider vicinity of the current search path. Other Database Algorithms dynamically adjust the Search Path based on the success of recent findings such as the detection of nodes that contain relevant contents or nodes whose authors hold relevant expertise.

Stage 3: Define an End Point (/s3)

The third group of Database Algorithms defines the End Point of a Search Path. The End Point may be determined based on an analysis of the node contents and properties, the total number of nodes examined, or the distance between the nodes and the Start Point. In other words, the placement of an End Point results from a state of satisfaction or increasing impatience and tiredness by the Database Algorithms. A typical example for this group of algorithms is the Patience Algorithm that terminates a search after the examination of a specified number of nodes.

Stage 4: Prioritize the Results (/s4)

The fourth group of Database Algorithms prioritizes the search results. Every database search returns the card referenced by the Start Point and all subsequently explored nodes. By default, the cards are returned in the order in which they were examined. Other criteria for prioritizing the search results include the distance between the nodes and the Start Point, the number of relations associated with nodes, the node contents, and the node properties such as card sizes, modification dates, authors, contents, notifications, comments, votes, accesses, font types, pictures, and locations.

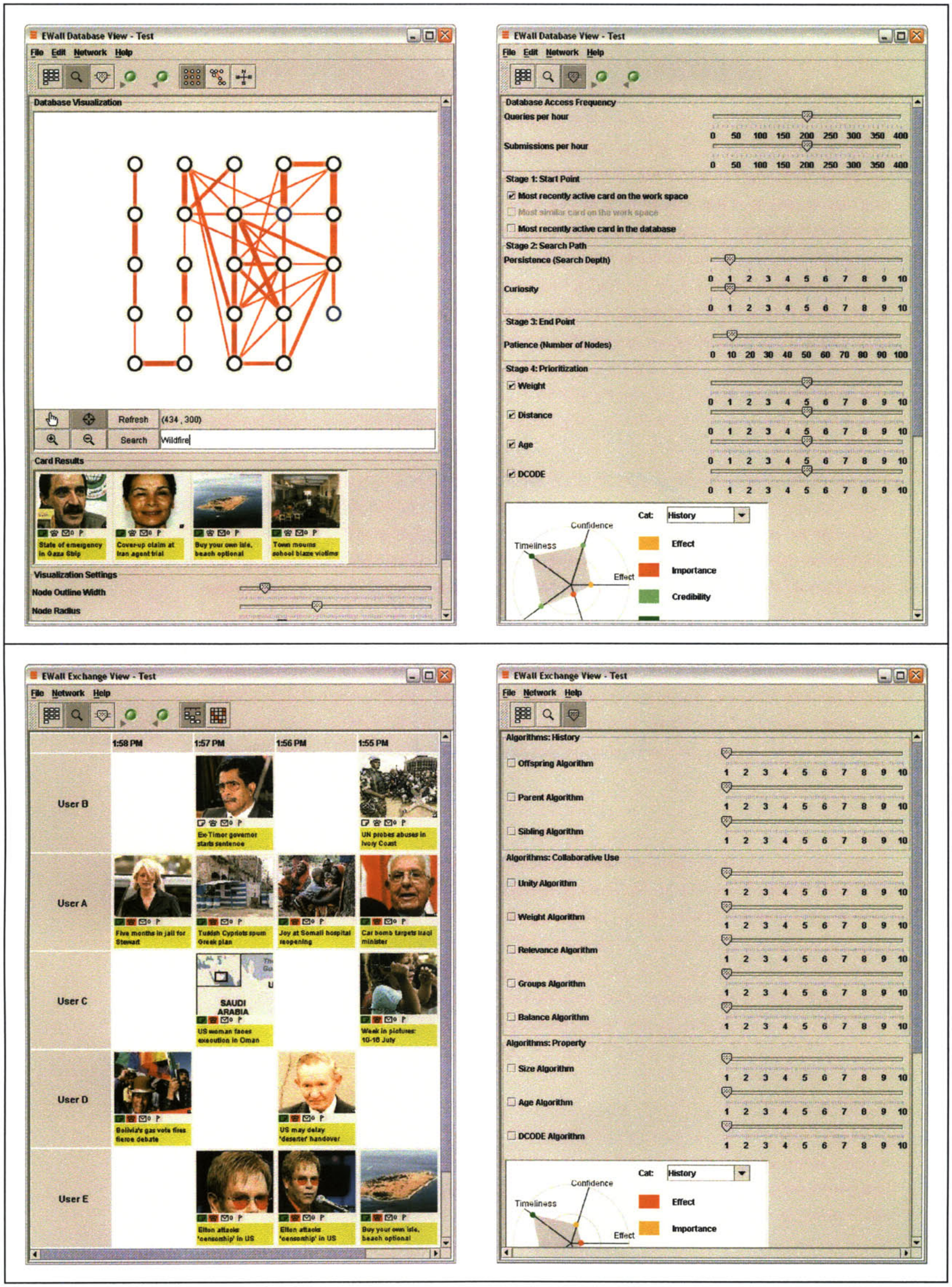


Illustration 5: User interfaces for the management of Database and Exchange Algorithms

2.2.1. Curiosity /s2

The Curiosity Algorithm is conceived as a means to consider less heavily connected nodes during database searches. By default, a search proceeds via the most heavily connected nodes. The Curiosity Algorithm fosters the exploration of less heavily connected nodes within a certain vicinity of the Search Path. For example, a Curiosity value of 1 would not only examine the adjacent node connected with the most relations but also nodes connected with one less relation (see Illustration 4). The exploration of less heavily related nodes can potentially alter the Search Path. For example, if nodes A, B and C are connected with two relations each, if nodes A and A1 are connected with one relation, and if nodes A1 and A2 are connected with three relations, then, given a Curiosity value of 1, the original search path would no longer proceed via node C but instead via node A2. The operation of the Curiosity Algorithm reflects the thinking of humans that not only consider what appears obvious but attempt to gain additional insight by critically investigating their thoughts, explore alternatives, and continuously reevaluate their objectives.

2.2.2. Persistence /s2

The Persistence Algorithm controls the search depth of explorations away from the current Search Path. For example, a Persistence Value of 1 would allow for the exploration of nodes that indirectly connect with the current Search Path via no more than one node (see Illustration 4). Heavily connected nodes often gain strength from less heavily connected nodes within their immediate vicinity. Thus, the analysis of such nodes can provide information of contextual relevance and uncover clues about the historic development of nodes and relations within specific parts of the network. The operation of the Persistence Algorithm displays similarities with humans that try to remember the past in order to understand and to reason concurrent knowledge.

2.2.3. Patience /s3

The Patience Algorithm terminates a search after the examination of a specified number of nodes. Illustration 4 shows the termination of a search after the exploration of 5, 10, and 15 nodes. The Patience Algorithm not only ensures that database searches only take place within domains of immediate interest but also prevents comprehensive and long lasting searches in large databases. The operation of the Patience Algorithm displays similarities with humans that mentally focus on a particular subject and that loose patience after failing to remember or discover relevant knowledge after an extended period of time.

2.2.4. Sequence /s4

The Sequence Algorithm prioritizes cards in the order in which they were discovered during a search process. Illustration 4 displays an example with 5 nodes that are sorted by the order in which they were examined. The Sequence Algorithm considers the search process as an exploration that leads into areas with nodes of increasingly less relevant contents. (This assumption may often prove untrue. However, every Algorithm has its individual opinion that remains influential until overshadowed by the opinions of other Algorithms.) The human analogy to this Algorithm is that initial thoughts about a particular subject are likely to influence or even supersede subsequent considerations.

2.2.5. Distance /s4

The Distance Algorithm prioritizes cards based on their distances from the Start Point. The distance is defined as the path with the smallest number of relations between a node and a Start Point. Illustration 4 displays an example with 9 nodes that are sorted by their distances from the Start Point. The Distance Algorithms assumes that only nodes in the vicinity of a Start Point are of relevance. The operation of the Distance Algorithm resembles the behavior of humans that explore and critically examine related knowledge that may support or contradict their preceding thoughts.

2.2.6. Weight /s4

The Weight Algorithm prioritizes cards based on their node weights. The node weight is defined as the number of relations associated with a node. For example, if node A has 2 relations with node B and 1 relation with node C then the weight of node A is 3. Illustration 4 displays an example with 13 nodes that are sorted by their weights. The Weight Algorithms assumes that heavily connected nodes are generally more important than less heavily connected nodes. Thus, the Weight Algorithms does not arrange cards by their relevance for a particular user or situation but by their overall importance. The human analogy to this Algorithm is the preference given to more frequently accessed knowledge.

2.3. Exchange Algorithms

Exchange Algorithms prioritize cards created by multiple users based on the history, collaborative use, and properties of cards. The prioritization is customized for every individual user and based on their particular interests and needs. The goal is to reduce information overload and to minimize the necessary amount of verbal and written communication among users during the exchange, evaluation, and comparison of cards.

Note: The use and operation of Exchange Algorithms is explained in Section A (Exchange Module).

H: Exchange Algorithms are designed based on how humans make sense of and appropriate information received from other people. Humans commonly evaluate information in terms of relevance, reliability, and importance. For example, humans might evaluate the relevance of information based on whether the information affects current activities or objectives, supports or confirms existing knowledge, or originates from a person with similar interests or backgrounds. Humans might evaluate the reliability of information based on whether the information provider is a specialist in the filed, has proven trustworthy in the past, has previously provided useful information, or offers information that does not contradict previously acquired knowledge. Humans might evaluate the importance of information based on whether the information receives the attention of many people or relates to issues of great consequence or urgency. Humans commonly refrain from a more careful evaluation of information if little time is available. For example, during a fire alarm people usually leave a building without confirming the alarm and without engaging in a more detailed investigation about the circumstances.

C: The primary function of the Exchange Algorithms is to determine the relevance of cards for individual users. The secondary function of the Exchange Algorithms is to determine the reliability and importance of cards. The Exchange Algorithms are divided into three groups (/h /c /p) that determine the relevance, reliability and importance of cards based on the history, the collaborative use and the properties of cards:

History of Cards (/h)

Exchange Algorithms with focus on the history of cards are conceived to decrease the relevance of cards that might contain redundant information. For example, a card could contain redundant information if it is a copy of another card and if neither the card original nor the card copy received major modifications. Every card has a unique and permanent identifier referred to as the Card ID. Every card copy receives a new Card ID yet maintains a record of all past Card ID's. Past Card ID's are referred to as Card History ID's. The comparison of Cards ID's and Card History ID's allows for the detection of cards that have the same heritage and that might contain redundant information.

Illustration 6 presents a short scenario that demonstrates how the card history might affect the prioritization of cards. The scenario shows the Workspace Views (white area) and the Exchange Views (gray area) of three users (UA-UC) over time (1-7). The Exchange Views display the cards on the Workspace Views of all collaborating users. The cards are arranged from left to right in the order of relevance. The relevance of individual cards is determined by the Sibling, Offspring, Parent, and Relative Algorithm (S, O, P, R). The illustration includes some visual features that are for explanatory purposes only and do not reflect the appearance of the actual user interface: First, the color red is used to highlight recent changes. Second, the small characters S, O, P, and R on top of cards indicate the particular Exchange Algorithm that currently affects a card's relevance. Third, some of the cards on the Exchange View are displayed on the far right even if space to the left is available. This allows for the easy recognition of cards with low relevance on an almost empty Exchange View.

The scenario starts with User A creating Card 1. Card 1 is displayed on the Exchange Views of all users (UA1, UB1, UC1). On User A's Exchange View the relevance of Card 1 is reduced because it is a Sibling (S) of a card User A is already using. User B copies Card 1 onto his Workspace View (UB2). User B can change the card size and location of Card 1 but User A remains in control of modifying the contents of Card 1. By taking ownership of Card 1 User B creates a new instance labeled Card 2. Card 2 is displayed on the Exchange Views of all users (UA3, UB3, UC3). On User A and B's Exchange Views the relevance of Card 1 is reduced because Card 1 is a Sibling (S) of a card User A is using and a Parent (P) of a card User B is using. Similarly, the relevance of Card 2

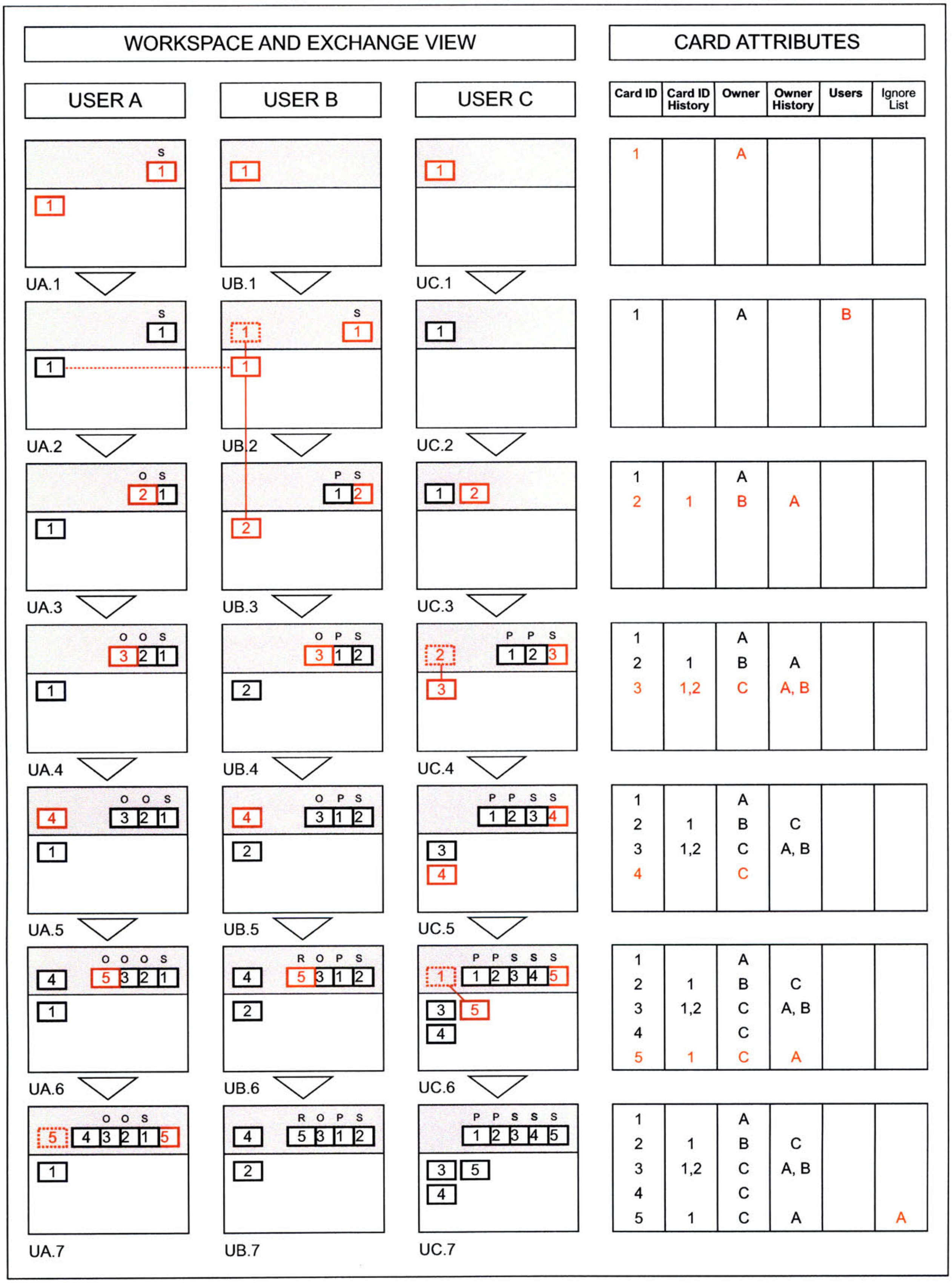


Illustration 6: Scenario / History of Cards

is reduced because Card 2 is an Offspring (O) of a card User A is using and a Sibling (S) of a card User B is using. User C copies and takes ownership of Card 1 (UC6) and Card 2 (UC4) thus creating Card 3 and 5. User C also creates Card 4 (UC5). Card 3, 4 and 5 are displayed on the Exchange Views of all users (UA6, UB6, UC6). For User A Card 5 is of low relevance because it is an Offspring (O) of Card 1, for User B Card 5 is of low relevance because it is a Relative (R) of Card 2, and for User C Card 5 is of low relevance because it is a Sibling (S) of Card 5. For User A and B Card 4 is of higher relevance because it is new to both users. Users can also manually decrease the relevance of cards on their Exchange Views. For example, after User A decides to decrease the relevance of Card 5 it becomes the last card displayed on his Exchange View (UA7).

Collaborative Use of Cards (/c)

Exchange Algorithms with focus on the collaborative use of cards increase the relevance of cards that might foster coherence among the foci, views, and objectives of collaborating users. For example, if a single user were to ignore a card that everybody else is currently working with then the Exchange Algorithms might increase the relevance of this card for this particular user.

Illustration 7a-c presents a short scenario of how the collaborative use of cards might affect the prioritization of cards: Illustration 7a shows the Workspace Views (white area) and the Exchange Views (gray area) of four users (UA-UD) over time (1-6). Illustration 7b shows the cards, relations, and collaborative information collected by the Recognition Functions of the Exchange Algorithms. Illustration 7c shows the prioritization (P1-4) of cards (1-14) for individual users (A-D) by the Execution Functions of the Exchange Algorithms. The following three paragraphs describe some of the events presented in Illustration 7a-c:

The scenario starts with User A and C creating Card 1 and 2 (UA1, UC1). The two new cards are displayed on the Exchange Views of all users except the card authors (UA1, UB1, UC1, UD1). User A and D copy the cards on their Exchange Views (UA2, UD2) to their Workspace Views. User B and D establish explicit relations among their cards: User B connects Card 3 and 4 with a rubber-line and User D groups Card 1, 2, and 5 with a bounding-box (UB3, UD3). User C enlarges the size of Card 6 thus increasing its visual significance (UC4). User A erases Card 2 (UA6). The order of cards on the Exchange Views of individual users differs and dynamically changes. For example, the order of Card 3 and 4 differs for User C and D (UC3, UD3) and the order of Card 10 and 11 changes for User A (UA5, UA6).

The contents on the Workspace Views and the activities of all users are monitored and recorded by the Recognition Functions of the Exchange Algorithms. The information collected by the Recognition Functions is organized and stored for the subsequent use by the Execution Functions and includes cards and relations as well as statistical information about user interactions and Workspace View contents. Recognition Functions with focus on Cards record the addition, modification and deletion of cards including the current and past card authors and card users. For example, RA1 indicates the addition of Card 1 and 2 by User A and C. RA2 shows that User D copied Card

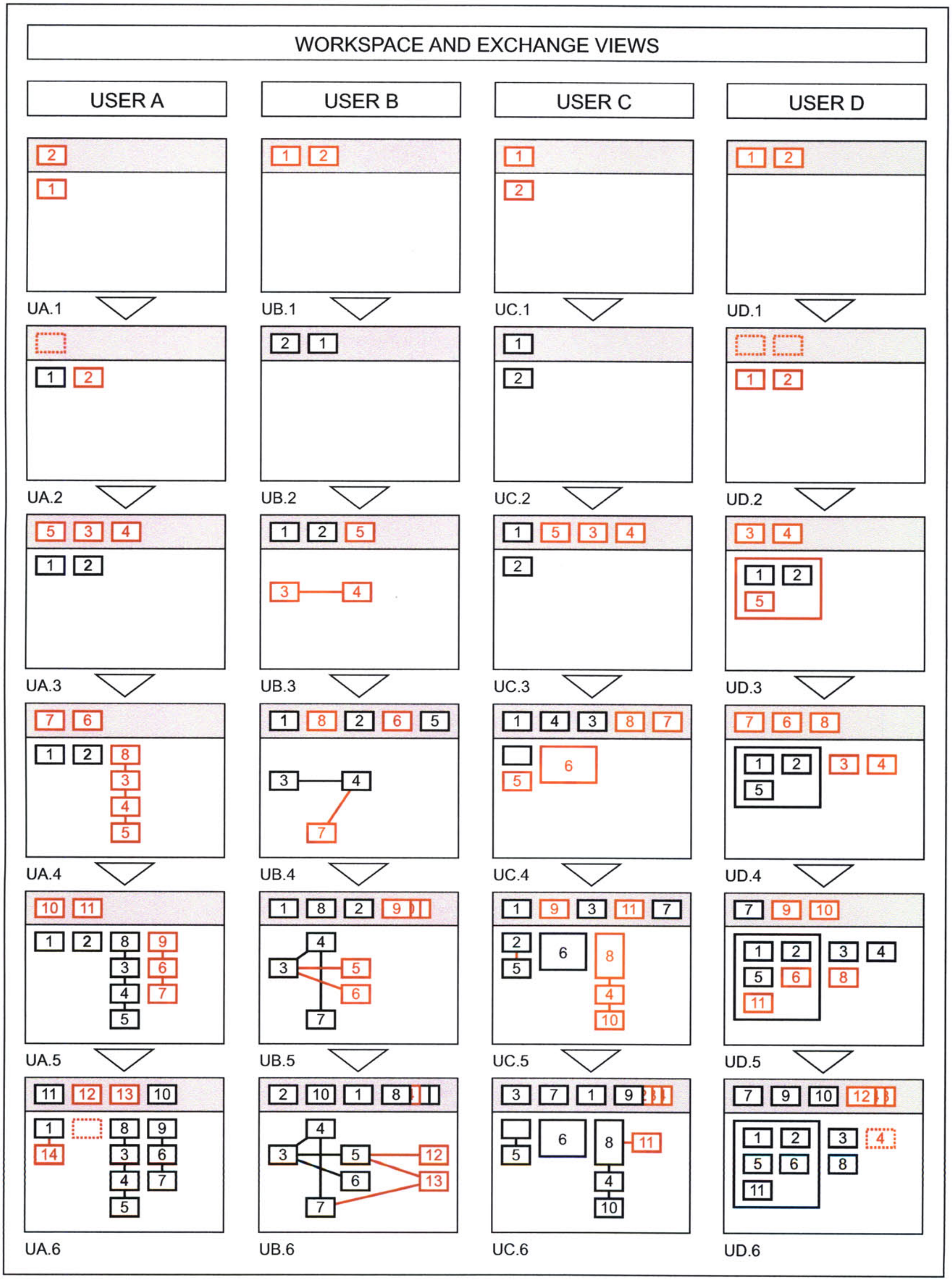


Illustration 7a: Scenario / Collaborative Use of Cards

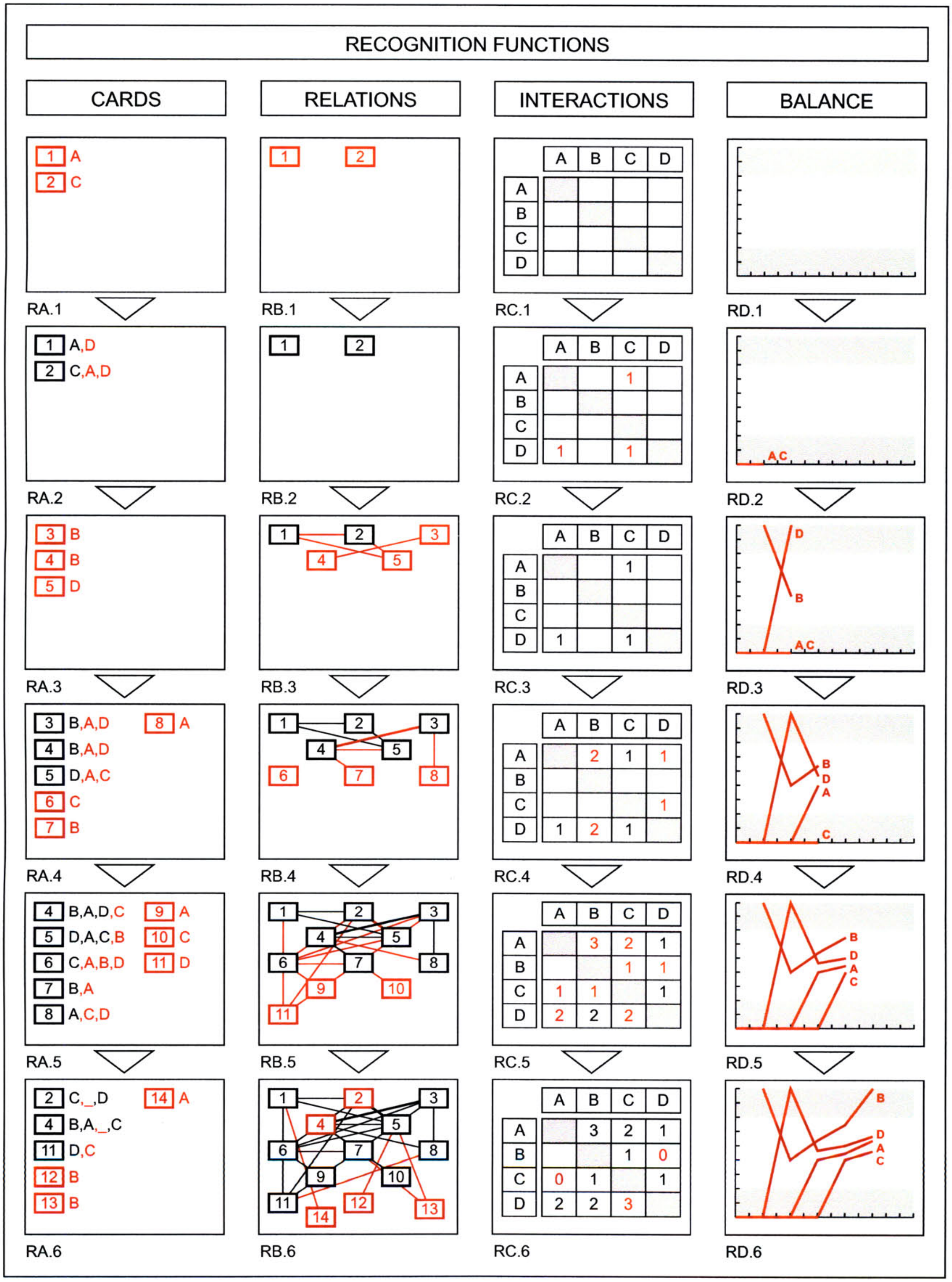


Illustration 7b: Scenario / Collaborative Use of Cards / Recognition Functions

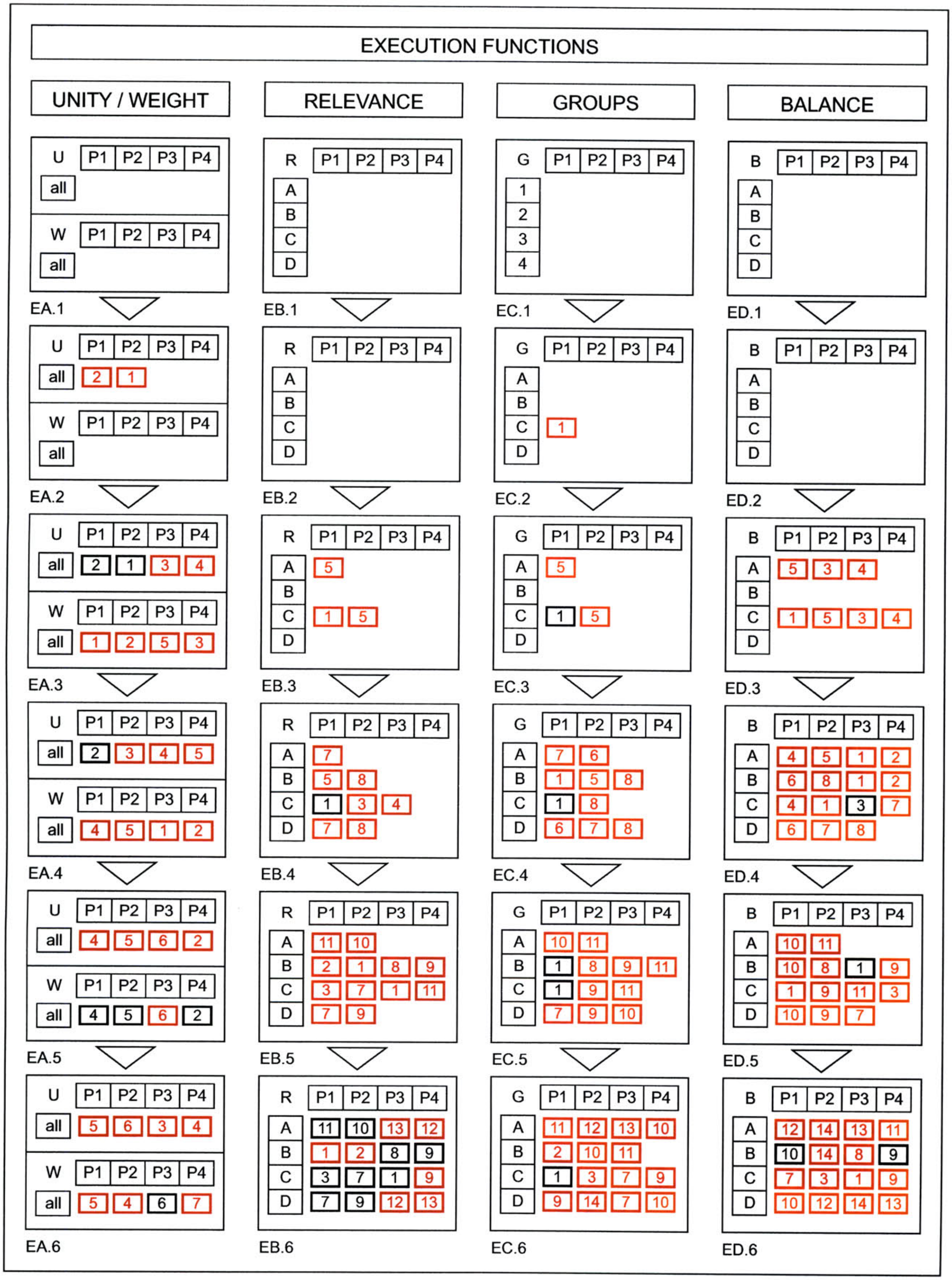


Illustration 7c: Scenario / Collaborative Use of Cards / Execution Functions

1 and that User A and D copied Card 2. Furthermore, RA6 shows that User A erased Card 2 and that User D erased Card 4. Recognition Functions with focus on Relations record and combine all explicit relations among cards established by users as well as all implicit relations established by Level I and II Algorithms. For example, RB3 shows the addition of an explicit relation between Card 3 and 4 established by User B through the use of a rubber-line (UB3) as well as the addition of three implicit relations among Cards 1, 2 and 5 established by User D through the use of a bounding-box (UD3). Recognition Functions with focus on Interactions keep track of all direct and indirect interactions among users. An interaction is registered if a user copies a card, views the contents associated with a card, or adds a comment, a vote or a notification to a card (see EWall Cards). The results are maintained in a table that lists the information providers along the y-axis and the information beneficiaries along the x-axis. For example, RC2 shows one interaction in cell A/C because User A copied Card 2 created by User C (UA2). Similarly, RC2 shows one interaction in cell D/A and one interaction in cell D/C because User D copied Card 1 from User A and Card 2 from User C (UD2). Recognition Functions with focus on Balance monitor the difference between the number of cards and relations on the Workspace Views of individual users. The graph displays the number of relations divided by the number of cards on the x-axis and time along the y-axis. For example, UB4 shows that User B works with three cards and two relations. Consequently, the current graph value for User B in RD4 is 2/3. Similarly, UA4 shows User A works with six cards and three relations. Thus, the current graph value for User A in RD4 is 3/6.

The Execution Functions of the Exchange Algorithms use the information collected by the Recognition Functions as a basis for prioritizing the cards on the Exchange Views of individual users. The Execution Function of the Unity Algorithm prioritizes cards by the number of current and past users. The results apply to all users. For example, EA2 shows Card 2 ahead of Card 1 because Card 2 was used by three users while Card 1 was only used by two users. The Execution Function of the Weight Algorithm prioritizes cards by the number of relations associated with cards. The results apply to all users. For example, EA3 shows Card 5 ahead of Card 3 because Card 5 is associated with two relations and Card 3 with only one relation (RB3). The Execution Function of the Relevance Algorithm also prioritizes cards by the number of relations associated with cards yet only considers relations that connect cards the viewer is using on his Workspace View. For example, EB5 shows Card 2 ahead of Card 1 for User B. RB5 shows that Card 1 is related with Card 2, 5 and 6, that Card 2 is related with Card 1, 4 and 6, and that Card 2 is related twice with Card 5. User B only uses three (4, 5, 6) of the five (1, 2, 4, 5, 6) cards on his Workspace View (UB5). Consequently, Card 2 has a total of four relations and Card 1 has a total of only two relations with cards that exist on User B's Workspace View. Thus, for User B Card 2 is considered more relevant than Card 1. The Execution Function of the Groups Algorithm prioritizes cards by the number of interactions the viewer had with the card authors. For example, EC6 shows Card 14 ahead of Card 7 for User D. Card 14 was created by User A and Card 7 was created by User B and later copied by User A. RC6 shows that User D and A had three interactions (A/D = 1 and D/A = 2) and that User D and B had only two interactions (B/D = 0 and D/B = 2). Thus, for User D User A's card (14) is considered more relevant than User B's card (7). The Execution Function of the Balance Algorithm prioritizes objects in ways that fosters a balance between the amount of cards and relations. For users with few cards and many relations this function proposes weakly related cards while for users with many cards and few relations this function proposes heavily related cards. For example, ED4 shows that for User C Card 4 is ahead of Card 1. UC4 shows that User C has not yet established any relations. (This example does not consider implicit relations established by Level II Algorithms.) As a consequence, the Execution Function increases the priority for heavily related cards. RB4 shows that Card 4 is related twice with Card 3 and once with Card 5 and 7 (a total of four relations). On the other hand, Card 1 is only related once with Card 2 and 5 (a total of two relations). Thus, for User C Card 4 is considered more relevant than Card 1.

The prioritized card sequences generated by the various Exchange Algorithms (EA, EB, EC, ED) are combined and displayed on the Exchange Views of individual users (UA, UB, UC, UD). (This scenario only introduces a basic method for the combination of card sequences and does not demonstrate the self-adjusting capabilities of the Exchange Algorithms.) Cards in Position 1 (P1) receive four points, cards in Position 2 (P2) receive three points, cards in Position 3 (P3) receive two points, and cards in Position 4 (P4) receive one point. The cards on the Exchange Views are arranged by their total number of points. For example, the order of cards on the Exchange View of User C (UC4) is 1 - 4 - 3 - 8 - 7 because Card 1 has 13 points (2 points in EA4 + 4 points in EB4 + 4 points in EC4 + 3 points in ED4), Card 4 has 10 points (4 points in EA4 + 2 points in EB4 + 4 points in ED4), Card 3 has 5 points (3 points in EB4 + 2 points in ED4), Card 8 has 3 points (3 points in EC4), and Card 7 has 1 point (in ED4).

Properties of Cards (/p)

Exchange Algorithms with focus on card properties are conceived to adjust the relevance of cards based on card sizes, modification dates, authors, contents, annotations, comments, votes, accesses, font types, font styles, font sizes, pictures, and location references. Object properties are often influential for how humans perceive, relate, and arrange objects. For example, stamp collectors often organize stamps by age whereas some collectors prefer arrangements that reflect the stamps' motives, values, or countries of origin. The evaluations of object properties also differ depending on the viewers and the circumstances. For example, car drivers are more likely to focus on smaller signs with abstract icons (that usually display driving instructions and warnings) as opposed to bigger signs with colorful graphics (that usually display advertisements).

2.3.1. Offspring /o

The Offspring Algorithm detects and adjusts the relevance of cards that are copies of cards the user himself created. For example, if User A copies (and takes ownership of) a card that User B created then the card copy is considered an Offspring and the card's relevance adjusted for User B. In other words, an Offspring is a card with a Card History ID that matches the Card ID of another card. Users can decrease the relevance of Offspring cards to maintain focus on information that is not based on their own work. Users can increase the relevance of Offspring cards to monitor the collaborative use of their contributions.

2.3.2. Parent /o

The Parent Algorithm detects and adjusts the relevance of cards that the user previously copied (and took ownership of). In other words, a Parent is a card with a Card ID that matches the Card History ID of another card. Users can decrease the relevance of Parent cards to maintain focus on information they do not already work with. Users can increase the relevance of Parent cards to monitor for modifications to cards they currently use copies of.

2.3.3. Sibling /o

The Sibling Algorithm detects and adjusts the relevance of cards that the user himself created. In other words, a Sibling is a card with a Card ID that matches the Card ID of another card. Users can decrease the relevance of Sibling cards to maintain focus on information they did not create themselves. Users can increase the relevance of Sibling cards to examine how their own contributions compare with and rank among the cards of other users.

2.3.4. Relative /o

The Relative Algorithm detects and adjusts the relevance of cards that have the same origins as cards the user is currently working with. For example, if User A and B copy (and take ownership of) a card that User C created then the two card copies are considered Relatives. In other words, a Relative is a card with a Card History ID that matches the Card History ID of another card. Users can decrease the relevance of Relative cards to maintain focus on information that is not same or similar as information they are already working with. Users can increase the relevance of Relative cards to detect cards whose contents may provide interesting variations to the contents of cards they are currently working with.

2.3.5. Unity /c

The Unity Algorithm adjusts the relevance of cards based on the number of current and past card users. A card user is anybody who created, modified, copied, commented, voted on, took ownership of, or reviewed the content of a card. The Unity Algorithm operates based on the assumption that the number of card users represents the popularity and subsequent relevance of cards.

2.3.6. Weight /c

The Weight Algorithm adjusts the relevance of cards based on the number of relations associated with cards. For example, if two relations connect Cards A and B and if one relation connects Cards B and C then the weight of Card A is 2, the weight of Card B is 3, and the weight of Card C is 1. The Weight Algorithm operates on the assumption that the card weights are representative for the access frequency and subsequent relevance of cards.

2.3.7. Relevance /c

The Relevance Algorithm adjusts the relevance of cards based on the number of relations associated with cards (same as the Weight Algorithm) yet only considers relations that involve cards the user is currently working with. For example, if a card is related with four other cards only two of which the user is currently working with then the card's weight is assumed 2 rather than 4. The Relevance Algorithm operates based on the assumption that users are particularly interested in information that specifically relates to what they are currently working on.

2.3.8. Group /c

The Group Algorithm adjusts the relevance of cards based on the number of interactions between the viewer and the card authors. An interaction is registered whenever a user copies, takes ownership of, adds a comment to, or votes for a card of another user. For example, if User A viewed the contents of two cards created by User B (two interactions between User A and B) and if User C copied one card created by User A (one interaction between User A and B) then, for User A, the relevance of User B's cards exceeds the relevance of User C's cards. The Group Algorithm also considers indirect interactions. For example, if one interaction occurred between User A and B as well as one interaction between User B and C then one indirect interaction is assumed between User A and C. The reasoning behind this is that User A could have shared information with User B that subsequently User B shared with User C or that User B could have shared the same information independently with User A and C. The Group Algorithms operates based on the assumption that the network of people interacting with each other indicates groups of people with potentially common foci, interests, or work tasks. The analysis of direct and indirect interactions not only enables the Group Algorithm to prioritize cards but also to provide users with the names of other users that may have similar interests, that may be experts on subjects of current consideration, or that may benefit from a direct collaboration. The objectives of the Group Algorithm are similar to Warren Zack's Conversation Map [52], a software application that examines and visualizes social networks, discussion themes and frequently use terms in newsgroup messages. The objectives also bear similarities with the PhaseX project [53], a software application that monitors and visualizes the evolution of user contributions during collaborative design exercises.

2.3.9. Balance /c

The Balance Algorithm adjusts the relevance of cards in ways that helps users to maintain a balance between the amount of cards and (explicit and implicit) relations among cards. The Balance Algorithm operates based on the assumption that few cards and many relations indicate a user focus on organizing information while many cards and few relations indicate a user focus on exploring information. While the exploration of information is mostly beneficial during the early stages of sense-making processes, the organization of information becomes increasingly critical during the later stages of sense-making processes. The Balance Algorithm adjusts the relevance of cards in ways that encourages users to maintain a balance between exploring and organizing information thus minimizing situations in which users get lost in details or loose focus. For example, if after an extended period of time a user operates with significantly more cards than relations then the Balance Algorithm would (for this user only) increase the relevance of heavily related cards.

2.3.10. Pattern /c

The Pattern Algorithm increases the relevance of cards in ways that fosters coherence among the work of individual users. The Pattern Algorithm searches for similarities among the card arrangements of different users and identifies cards that could complete similar card patterns. For example, if User A horizontally aligns Card 1, 2, 3, and 4 and if User B horizontally aligns Card 1, 3, and 4 then the Pattern Algorithm would increase the relevance of Card 2 for User B.

2.3.11. Popularity /c

The Popularity Algorithm increases the relevance of cards created by popular users. A popular user is a user whose past contributions have been adapted by many other users. To prevent a permanent advantage of popular users (Monopoly) the Popularity Algorithm only considers recent contributions. The Popularity Algorithm emulates a typical aspect of conventional meetings where contributions by respected and successful participants are more likely to receive attention and consideration.

2.3.12. Focus /c

The Focus Algorithm increases the relevance of recently active cards. Card activities are registered whenever a user selects a card such as when viewing, moving, or editing cards. The purpose of the Focus Algorithm is to encourage remotely distributed collaborators to develop a coherent focus or at least to become aware of their individual foci.

2.4. Visualization Algorithms

Visualization Algorithms analyze and visualize networked information structures. The primary purpose of the Visualization Algorithms is to provide users with a variety of options for viewing cards on their News, Database and Exchange Views. Every Visualization Algorithm is focused on the analysis and visualization of one particular aspect of a networked information structure (one-dimensional visualizations) though multiple Visualization Algorithms can be combined to consider multiple aspects simultaneously (multi- dimensional visualizations).

Note: The use and operation of Visualization Algorithms is explained in Section A (Visualization Module).

H: Visualization Algorithms are designed based on how humans establish, recognize and mentally comprehend relations among virtual and physical objects such as documents, files or books. Humans organize and reorganize objects to study, preserve and communicate relations among objects. For example, humans arrange documents, files and books in ways that allow themselves and other people to quickly and easily find what they are looking for. Although different individuals have different ways of organizing objects, individuals with similar backgrounds that operate under similar circumstances often organize objects in similar ways. While the differences in how humans create and perceive organizations of objects may cause confusion they also instigate constructive dialogues and foster new, alternative and in-depth interpretations.

C: Visualization Algorithms analyze and visualize cards in ways that help users recognize, comprehend and leverage probable associations among cards. Users can apply and combine Visualization Algorithms to highlight and compare such associations. Visualization Algorithms are divided into three groups: Visualization Algorithms with focus on Spatial Arrangements (/s) spatially arrange cards in ways that reflect the types and numbers of relations among cards. For example, two cards connected with many relations might be located near each other. Visualization Algorithms with focus on Visual Appearance (/v) supplement spatial card arrangements with additional information about the types and numbers of relations among cards. For example, cards connected with few relations might be faded. Visualization Algorithms with focus on Analysis (/a) do not display cards but provide statistical information about the types and numbers of relations among cards. For example, a graph might indicate the total number of relations among cards over time.

2.4.1. Random /s

The Random Algorithm orthogonally aligns a random sequence of cards. The random arrangement of cards is used as a default arrangement to inspire an objective exploration of cards. The random arrangement is also used to evaluate the effectiveness of other types of card arrangements. (The effectiveness of a card arrangement is determined based on how fast a user can detect relevant information and compared to the effectiveness of a random card arrangement.)

2.4.2. Circle /s

The Circle Algorithm arranges cards in a circle. This arrangement is primarily used for the examination of relations. The advantage of a circular card arrangement is that relations between different pairs of cards do not overlap and that the number of realtions associated with individual cards becomes easily recognizable. The Circle Algorithm also displays a Center of Gravity among cards. (The Center of Gravity refers to a spatial location that balances the weights of all cards. The card weights reflect the number of relations associated with cards. In a circular card arrangement the cards closest to the Center of Gravity have the highest weights.) Users can move the Center of Gravity to relocate cards relative to the card weights.

2.4.3. Hierarchy /s

The Hierarchy Algorithm arranges cards hierarchically using as its top a card selected by the user. The particular hierarchical representation for this arrangement is modeled after an outline and allows users to expand and collapse individual branches. Because

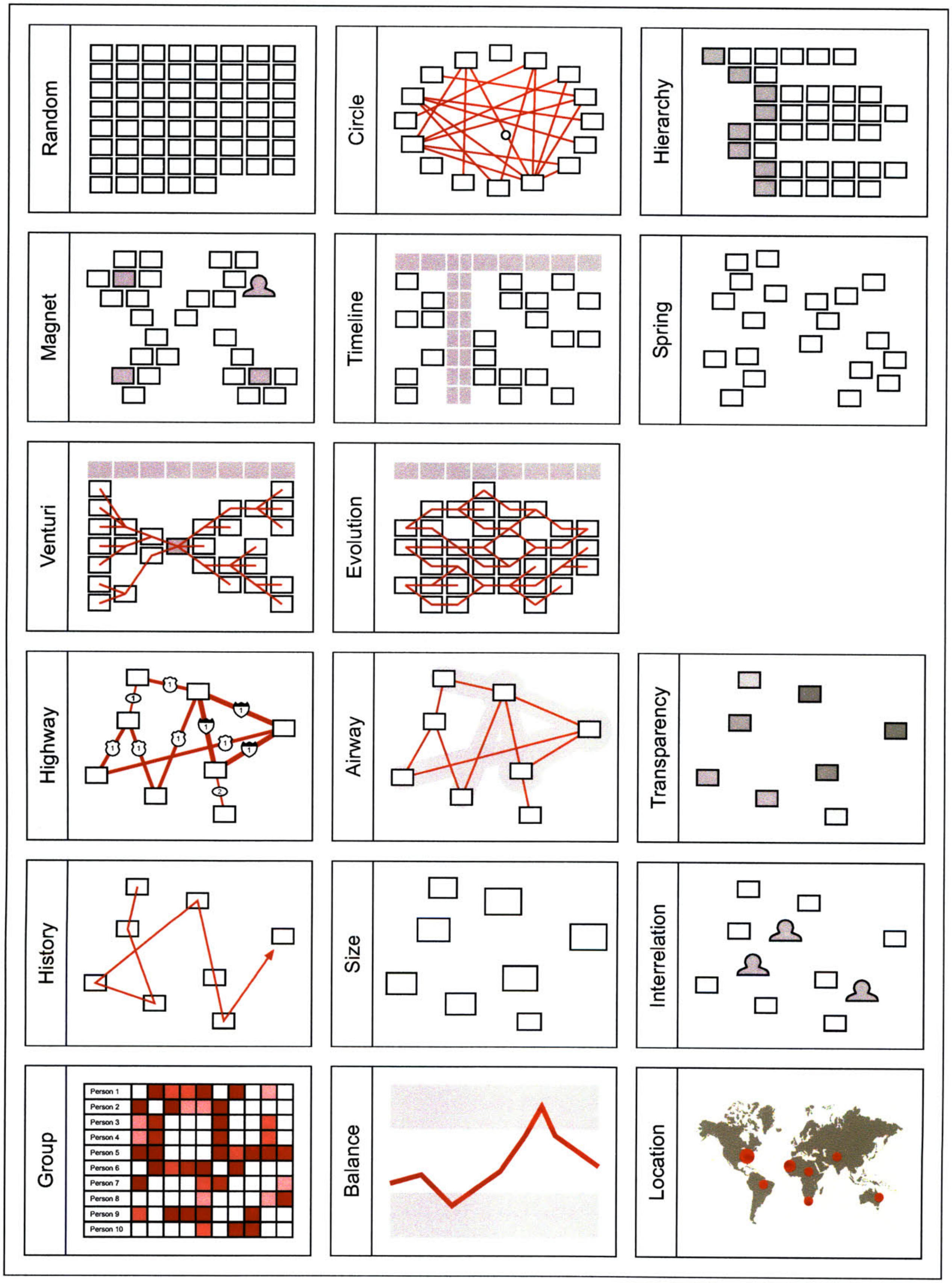


Illustration 8: Visualizations

the hierarchical representation of networked structures might produce redundant information (e.g. a network that connects nodes A - B - C - A - D could produce an infinitely large hierarchical structure with branches such as A - B - C - A - B - C ...) this algorithm terminates individual branches whenever a node is discovered for the second time (e.g. the previous example would result in a hierarchy with branches A - B - C - A and A - D). Hierarchies are among the most familiar ways of organizing information and computational software for the hierarchical visualization of networked information is widely available. Particularly successful examples include inxight Star Tree [54], The Brain [55], WebMap [56], and Kart00 [57].

2.4.4. Magnet /s

The Magnet Algorithm arranges cards relative to two or more cards selected by the user. The distances between the selected and unselected cards reflect the number of relations between the cards. For example, if two relations were to connect cards A and B, if one relation were to connect cards B and C, and if cards A and C were currently selected, then the Magnet Algorithm would position card B in a distance from card C that is twice the distance from card A. The Magnet Algorithm is primarily used to explore cards that are related to cards of present interest. VisualWho [58] provides an example of a visualization based on similar principles.

2.4.5. Timeline /s

The Timeline Algorithm arranges cards in a table. The cards are organized by modification dates or references to time-based events in horizontal direction and by categories or information sources in vertical direction. The horizontal and vertical scales are optimized to display as many cards as possible. New cards are added to the left-most column on the designated category/source row. If the cell is already occupied then a new column is created and existing columns are shifted to the right. Thus, the time scale in horizontal direction is irregular. More specifically, every column covers the time period between its most recent addition and the most recent addition to its preceding column. The order and the heights of rows automatically adjust depending on the number of cards displayed in each row. For example, rows with many cards expand while rows with no cards are not displayed at all. The card arrangement produced by the Timeline Algorithm has proven effective for both continuous monitoring and occasional browsing of new information. Users can easily recognize and compare the chronologies of additions and modifications to different category/source rows.

2.4.6. Spring /s

The Spring Algorithm arranges cards so that the distances between cards reflect the number of relations between cards. For example, heavily related cards might be positioned close to each other while weakly related cards might be positioned further apart (or vice versa). This arrangement allows users to study associations among cards without the display of relations. This particular way of arranging information has been widely researched (Defining Digital Space through a Visual Language [59], Graph Layout Toolkit [60], Spring Embedding Algorithm [61], WebSom [62]) and commercially utilized (SmartMoney [63], Map.net [64], ThemeScape [65], City'O'Scope [66]).

2.4.7. Venturi /s

The Venturi Algorithm chronologically arranges cards relative to one card selected by the user. Related cards with more recent modification dates are positioned to the left and related cards with less recent modification dates to the right of the selected card. To allow for a larger number of cards to be displayed simultaneously, individual columns display multiple cards if they have similar modification dates and if they do not relate to each other but to cards in one or both adjacent columns. This particular visualization allows users to study the incremental development and dependencies of ideas and information. For example, cards displayed on the right might have inspired the creation of the selected card while cards displayed on the left might have been created in reference to the selected card.

2.4.8. Evolution /s

The Evolution Algorithm is similar to the Venturi Algorithm yet does not focus on a user specified card but a user specified time frame. The Evolution Algorithm is used to study the evolution, convergence, and separation of simultaneously evolving information and threads of ideas such as the convergence of multiple ideas into one solution or the conception of multiple alternative ideas based on an existing idea.

2.4.9. Highway /v

The Highway Algorithm visualizes associations with multicolored lines and symbols similar to the ones used on road maps. Different line types represent different numbers of relations between cards. A sequence of nodes connected with the same line types is referred to as a road. Every line type is associated with a unique symbol that displays the road number. The Highway Algorithm helps users to explore potentially meaningful pathways across the network of cards. The Highway Algorithm is best used in combination with the Spring Algorithm as it reduces the number of crossovers between lines (fewer bridges across roads), increases the line length for strong associations (long, large and busy roads between big cities), and decreases the line length for weak associations (short, small and infrequently accessed roads in residential areas).

2.4.10. Airway /v

The Airway Algorithm helps users detect heavily connected network segments by highlighting strong nodes and associations with big circular and rectangular shapes of gray color. The shapes produce patterns similar to the ones created by airport control areas and airway corridors on flight maps. The Airway Algorithm renders the shapes behind other visual components and thus can be used in combination with most Visualization Algorithms.

2.4.11. Transparency /v

The Transparency Algorithm represents the values of a specified card attribute with different levels of card transparencies. For example, card transparencies based on modification dates help users focus on more recent information without completely ignoring past content. Similarly, card transparencies based on weights (numbers of relations) help users detect unrelated cards that might require reconsideration or removal.

2.4.12. History /v

The History Algorithm visualizes the chronology of card additions and modifications with red arrows. The red arrows allow users to review the development of a card arrangement without having to compare multiple previously saved versions of the card arrangement. The red arrows also allow users to contrast the development of a card arrangement with aspects highlighted by other Visualization Algorithms.

2.4.13. Size /v

The Size Algorithm represents the values of a specified card attribute with different card sizes. The Size Algorithm allows users to reduce the sizes of cards that are not of current focus or interest. The simultaneous application of the Spring Algorithm further optimizes the available display area by minimizing the spacing between cards. (Axel Killian's work on "Defining Digital Space through a Visual Language" [59] introduces a more advanced visualization based on similar principles.)

2.4.14. Interrelation /v

The Interrelation Algorithm complements card arrangements with symbols that represent card attribute values such as particular modification dates, author and user names, or geographic locations. The individual symbols are located in the center and sized relative to the number of all cards with corresponding attribute values. For example, a symbol that represents a particular person is placed in the center of all cards created, modified, or used by this person. The Interrelation Algorithm is conceived to increase situational awareness in collaborative settings as well as to highlight possible correlations between the arrangement of cards and the card attributes.

2.4.15. Group /a

The Group Algorithm visualizes the total number of interactions between users (or the total number of relations between cards) in a table. (An interaction is registered whenever a user copies, takes ownership of, adds a comment to, or votes for a card of another user as well as for every relation that is created between the cards of two different users.) The table row and column headings display the names of users (or headings of cards) and are ordered so that frequently interacting users (or heavily related cards) are located near each other. The table cells indicate, with different shades of red color, the number of interactions between users (or the number of relations between cards). The Group Algorithm helps users to determine possible team arrangements (or card groupings) during collaborative sense-making activities.

2.4.16. Balance /a

The Balance Algorithm displays with a line graph the difference between the total number of cards and relations over time. A small number of relations and many cards indicate a less structured and more arbitrary arrangement of cards. A large number of relations and few cards indicate a well organized and structured arrangement of cards. An unstructured arrangement of cards is more typical during the beginning of a sense-making session when people primarily focus on exploring and collecting information. A structured arrangement of cards is more typical during the end of a sense-making session when people conclude their analyses and organization of information. The Balance Algorithm may be used to monitor progress during sense-making sessions as well as to recognize situations that allow for a broader exploration of issues or situations that suggest a more focused investigation.

2.4.17. Location /a

The Location Algorithm displays, with red symbols on a map, the geographic locations associated with cards. Cards with similar geographic locations do not produce multiple symbols but bigger symbols. The symbol sizes are proportional to the number of cards they represent. The Location Algorithm helps users monitor and separate location specific information. For example, the Location Algorithm may be used to display on a world map the geographic locations associated with recent newspaper articles.

Implementation Status and Credits for Section B

The Algorithms already implemented include 1.2.1, 1.2.5, 1.2.10, 1.3.1, 2.2.1, 2.2.2, 2.2.3, 2.2.4, 2.2.6, 2.3.1, 2.3.2, 2.3.3, 2.3.5, 2.3.6, 2.3.7, 2.3.8, 2.3.9, 2.4.2, 2.4.5, 2.4.6, 2.4.9, 2.4.10, 2.4.11, 2.4.13, 2.4.14, 2.14.15, 2.4.16, and 2.4.17. The Algorithms were conceived by Paul Keel and implemented by Michael Kahan, Yao Li, Raudel Rodriguez, Mathew Sither, and Benjamin Wang. The development of the Algorithms progressed under the consultation of Edith Ackerman, Jeffrey Huang, William Porter, and Patrick Winston.

SUMMARY

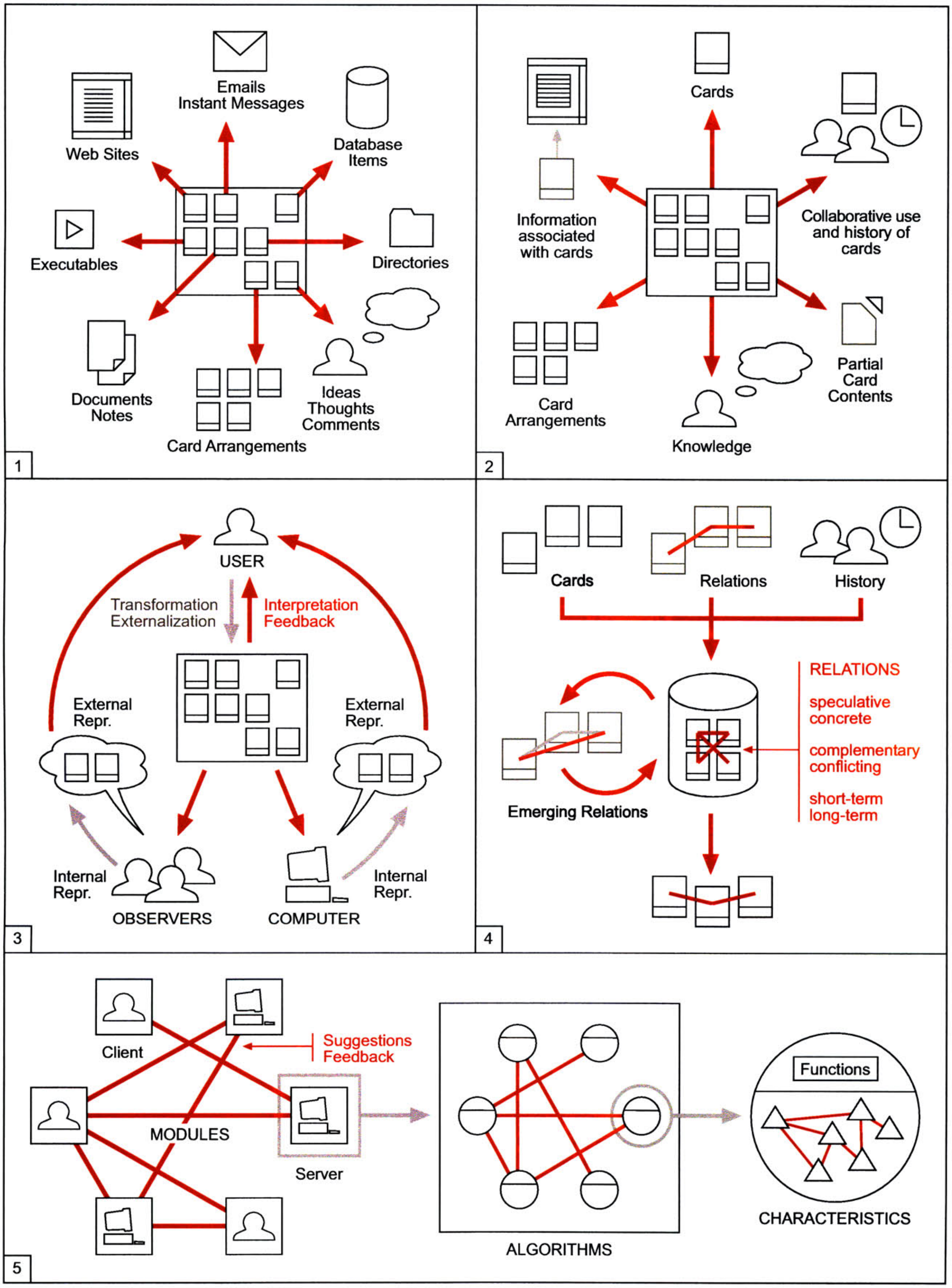
This work introduces a software application for the support of individual and collaborative sense-making activities. The software application is centered on five concepts for thinking about, managing and computationally processing data, information and knowledge:

1. Object Oriented Thinking (Cards)

The software application allows users to encapsulate, organize and compare data, information and knowledge through the use of card-like objects (EWall Cards) on a computer canvas (EWall Workspace View). The cards are conceived as a potential replacement for desktop icons. The software application includes functionality to create, modify, remove and arrange cards as well as to combine, separate and substitute card contents. To think of data, information and knowledge in terms of objects offers several benefits: First, the use of cards allows for the easy comparison of different data, information and knowledge formats such as ideas, comments, documents, notes, executables, web sites, emails, instant messages, database items, files, and directories. Second, the creation of cards encourages users to carefully evaluate and abstract data, information and knowledge. Third, cards convert data, information and knowledge into something more tangible that can be possessed, collected, and traded. Fourth, cards allow for complementary functionality such as attaching files and executables, adding comments and annotations, or hyper-linking remote contents and card arrangements.

2. Emergent Associations (Relations)

The software application detects explicit and implicit relations among cards. Explicit relations are usually more obvious and often interpreted similarly by different people. Examples of explicit relations include relations among spatially arranged cards grouped with bounding boxes or connected with rubber lines. Implicit relations are usually less obvious and often interpreted differently by different people. Examples of implicit relations include relations among spatially arranged cards in close proximity or cards created during similar time frames. Relations not only reflect associations among entire cards but also partial card contents, card histories, data and information associated with cards, or human knowledge stimulated by cards. For example, a relation between two cards could materialize if the first card contains information on a subject that the author of the second card is an expert on. Any collection of cards can contain a virtually unlimited number of possible relations. Furthermore, different relations emerge and dynamically change depending on the viewers' interpretations and the contextual circumstances. The software application identifies probable relations based on the spatial arrangement, the history, and the collaborative use of cards. The software application is open to future additions that will also allow for the examination of data and information associated with cards, partial card contents, and computational knowledge (previously accumulated cards and relations) stimulated by cards.



Emails
Instant Messages
Web Sites
Database
Items
Executables
Directories
Documents
Notes
Ideas
Thoughts
Comments
Card Arrangements
1
Cards
Information
associated
with cards
Collaborative use
and history of
cards
Partial
Card
Contents
Card
Arrangements
Knowledge
2
USER
Transformation
Externalization
Interpretation
Feedback
External
Repr.
External
Repr.
Internal
Repr.
OBSERVERS
COMPUTER
Internal
Repr.
3
Cards
Relations
History
RELATIONS
speculative
concrete
complementary
conflicting
short-term
long-term
Emerging Relations
4
Client
Suggestions
Feedback
MODULES
Server
Functions
ALGORITHMS
CHARACTERISTICS
5

Concepts

3. Circular Information Flow (Views)

The software application provides individual users with a prioritized selection of information that it considers relevant to their work and that might stimulate future additions and modifications to their card arrangements. The software application is analogous to, and acts in parallel with, human observers that review and respond to a user's card arrangement. The user and the software application communicate indirectly meaning that the user gathers information through the evaluation of suggested additions by the software application and the software application gathers information through the analysis of the user's evolving card arrangement. This "circular information flow" suggests a process by which the user and the software application make each other more "knowledgeable". The "circular information flow" leverages and brings together the human capacity for intuitive problem solving and the computer's capability for processing large amounts of information. Furthermore, the "circular information flow" causes minimal interference with human sense-making activities as it does not impose a particular work process, does not depend on direct human-computer interaction, does not confront humans with internal computational analyses, and clearly separates the interfaces controlled by humans (EWall Workspace Views) and computers (EWall News, Database, and Exchange Views).

4. Dynamic Knowledge Construction (Databases)

The software application collects, combines and structures information from different users and information sources. This information includes cards, explicit and implicit relations among cards, as well as information about the history and collaborative use of cards and card arrangements. All of the information is accumulated in a database with a structure that evolves and dynamically changes through the continuous addition of cards and relations. More than one relation can exist between any two cards thus allowing for an infinite number of possible network configurations. A large number of relations between two cards indicates a more concrete association while a small number of relations indicates a more speculative association. Relations can be temporary (dynamic / short term memory) or permanent (static / long term memory). Cards and permanent relations cannot be removed from the database yet the significance of associations among cards can diminish through an associative increase among other cards. Because all cards in the database interconnect directly or indirectly, the addition of every new relation can impact the balance of the entire network. Relations reflect the complementary and conflicting opinions of different algorithms. While some relations are established by algorithms on the client side (Level I and II Algorithms), others are created by algorithms on the database side (Level III and IV Algorithms): Algorithms on the client side establish relations based on the users' organization of cards. Algorithms on the database side establish relations based on the history and collaborative use of cards as well as on the databases' organization of previously accumulated cards and relations.

The database mechanics bear similarities with various mental processes: A first example is that associations between cards in the database evolve and dynamically change over time. The human analogy is that any interpretation of knowledge is unique. In other words, humans never produce the precisely same thought twice.

This is primarily due to a human's continuously evolving knowledge structure as well as a human's changing foci, needs, and objectives. While these inconsistent interpretations of human knowledge may be viewed as limitations they are essential for the creative conception of new ideas. A second example is that associations between cards become more resistant to change with the growing number of relations in the database. The human analogy is that children are commonly more flexible in creating and adapting new knowledge since they are less constrained and influenced by previously acquired knowledge. On the other hand, adults are more likely to hold on to their views and develop a tendency to accumulate new knowledge that does not fundamentally contradict previously acquired knowledge. While the decreasing adaptability of the human mind may be viewed as a limitation it is what makes humans unique in terms of style and character.

5. Decentralized Information Exchange (Modules, Algorithms, Characteristics)

The software application is divided into several components and sub-components. Components are autonomous and decentralized entities that have different functions and objectives, exchange information, and in some cases, negotiate their conclusions. Individual components can be added, removed, customized, and interconnected; thus providing users with a flexible and scalable computational environment for various collaborative settings and work tasks. The top-level components are referred to as Modules and consist of Client and Server Applications. Users operate Client Applications to exchange information with Server Applications and with the Client Applications of other users. Multiple Server Applications may be interconnected to join communities of users and to share resources. Every Client and Server Application contains multiple Algorithms, each of which consists of a Recognition and an Execution Function. Recognition Functions detect and retrieve information that Execution Functions need to create and analyze relations. Algorithms also contain human-like Characteristics (such as competitiveness, fatigue, and self-confidence) that control the exchange and negotiation of information. Characteristics change their states based on user feedback as well as based on the states of other Characteristics from the same or from different Algorithms. For example, a positive user feedback might increase an Algorithm's self-confidence, an increase in an Algorithm's fatigue might lower its competitiveness, and an increase in an Algorithm's competitiveness might increase another Algorithm's fatigue.

CONTRIBUTIONS

This work expands on the philosophy, conceptualization, and implementation of a modular computational environment (EWall Application) for the support of individual and collaborative sense-making activities. Particular contributions include:

the conceptualization and implementation of a customizable information format (EWall Cards) for the encapsulation of data, information and knowledge. The current implementation demonstrates the creation, customization, and operation of EWall Cards.

the conceptualization and implementation of five interoperable software components (EWall Modules) for the management of EWall Cards. The current implementation demonstrates the use of the individual Modules to collect, organize, and comprehend information (Workspace Module), to monitor for additions and modifications to different information sources (News Module), to combine and structure information from multiple users and information sources (Database Module), to prioritize the information exchange among users (Exchange Module), and to analyze and visualize information (Visualization Module). The current implementation also demonstrates how the combined use of the individual Modules enables a computational process (Circular Information Flow) by which the users and the EWall Application collaboratively develop an understanding of a particular situation and by which the EWall Application becomes adaptive to changing users and circumstances.

the identification of several combinations of EWall components (EWall Settings) for the support of specific work tasks and collaborative settings. The possibility for users to dynamically add, remove and recombine EWall components demonstrates the flexibility and scalability of the EWall Application.

the conceptualization and partial implementation of a database (Semi-structured Data Space) with a structure that evolves and dynamically changes through the accumulation of EWall Cards and relations, the analysis of the history and collaborative use of EWall Cards, and the examination of previously accumulated EWall Cards and relations. The current implementation demonstrates the unsupervised collection, combination, and organization of contents from multiple users and information sources.

the identification, conceptualization and partial implementation of about 25 computational mechanisms (Interpretation Algorithms) for the creation of relations among EWall Cards as well as about 35 computational mechanisms (Transformation Algorithms) for the retrieval, prioritization, and visualization of EWall Cards. The computational mechanisms demonstrate how theories and observations of human cognitive processes can be translated into discrete and interoperable computational algorithms and how these algorithms can be used to support human sense-making activities.

APPENDIX A DATA SPACES

The proposed algorithms are designed for the use with semi-structured data spaces that operate with data items and relations between data items. The distinction between structured, unstructured and semi-structured data spaces is essential for understanding the purpose and mechanics of the proposed algorithms. In general, structured data spaces contain no ambiguity in terms of data relations, unstructured data spaces contain no data relations at all, and semi-structured data spaces deal with dynamic and partially ambiguous data relations. This appendix compares the three different types of data spaces based on the simplicity of setting up and changing a data structure (Structure), the ambiguity and dynamics of relations between data items (Context), and the simplicity of adding and retrieving data items (Access).

Data-Space	Structure		Context		Content	
	Setup	Changes	Ambiguity	Dynamics	Additions	Retrieval
Structured	complex	complex	low	low	slow	fast
Semi-Structured	n/a	n/a	some	some	medium	medium
Unstructured	n/a	n/a	high	high	fast	slow

1. Structured Data Space:

A structured data space refers to conventional database applications where data is either hierarchically organized or otherwise categorized. For example, the words "red", "green", and "blue" may be assigned to the category "color" or made hierarchically subordinate to the word "color". Data members can belong to multiple categories but usually only reside in one place within a hierarchical structure.

Structure: A structured data space requires careful planning as later modifications of the data structure might turn out to be time intensive and complicated. Some of the planning issues concern the design of a template for individual data members and the design of valid categories. A comprehensive list of data categories will cover a wide range of possible data members. However, the more comprehensive a list of data categories becomes the more time it takes to set up the data space, to find the right category for each individual data member, to change the categories at a later stage or to search for a specific data member. A narrow list of categories on the other hand may be very limiting in terms of fitting new data members into existing categories often resulting in a miss-categorization and consequent loss of data or the forced alteration of data members for the compliance with existing categories.

Context: A structured data space is commonly static meaning that the data relationships and the data structure as a whole is unchanging. This structural persistence of data spaces is advantageous where the correct categorization of data items is unambiguous and permanent. However, if the categorization of data items frequently changes due to circumstances, user opinions and data actuality then the maintenance of the data space becomes inefficient.

Access: Searching a structured data space is efficient and produces concise results provided one knows what to look for. Random explorations of structured data spaces or the attempt to find similar or related information for a specific data member is less effective. The time it takes to add and retrieve data members increases with the amount of data and the complexity of the data structure.

2. **Unstructured Data Space:**

An unstructured data space refers to a data space that does not offer any information about potential relations between data items. In an unstructured data space all data items are stored in the same location. The absence of contextual information allows user to freely assume relations between data items.

Structure: An unstructured data space does not require planning and might prove effective for situations in which the future use and structure of the data is yet to be determined. Typical examples include data collected during brainstorming sessions or data accumulated during extensive library and web searches. Unstructured data spaces often constitute temporary solutions ahead of a more detailed investigation into appropriate data structures.

Context: Potential contextual relations in unstructured data spaces are only established through the creative interpretations of individuals. These temporary interpretations may reflect changing circumstances and differing backgrounds of individuals. The dynamics and ambiguity of such relations allow for a wide range of readings to emerge.

Access: The data access of an unstructured data space becomes increasingly difficult with the growing amount of information. To retrieve a specific data member one has to examine the entire data space. Computational systems effectively support this process by examining every individual data member for user specified key words (full-text search).

3. **Semi-structured Data Space:**

A semi-structured data space combines some of the advantages and disadvantages of both, structured and unstructured data spaces. The goal is to maintain a data space that provides enough structure to efficiently access data items while at the same time being adaptive to structural changes and contextual interpretations.

A successful example of a semi-structured data space is the World Wide Web. This heterogeneous and decentralized data space does not require data items to have a specific format nor does it require data items to be saved in a specific place. Data items are not categorized into groups or hierarchies nor are keywords required. Hyperlinks are the only means of structuring data items and common ways of retrieving data items are to follow hyperlinks or to explore the data content with search engines. On the www the community of content providers decides about data actuality.

Less popular or outdated web sites are typically unlinked and therefore structurally detached from the WWW; popular and new sites are being advertised and added to the WWW. The WWW provides us with functionality and a philosophy that is novel to conventional database systems. It proves to be especially useful in cases where the best possible categorization for the data is unknown, where the data actuality is temporally limited and where the reading of the data should be flexible and inspiring.

Structure: The structure of semi-structured data spaces is not predefined but continuously evolving through the additions of data items and relations by individual content providers. The motivation lies in creating data spaces without current knowledge about its future use. Data items are interlinked rather than categorized or hierarchically organized.

Context: A semi-structured data space is contextually defined through the relations between data items. These relations are established through content providers or computational mechanisms. Content providers establish relations between data items by complementing individual data items with pointers that reference other data items. These pointers are equivalent to hyper-links on the WWW and usually one-directional meaning that data items that contain the pointers are active while the referenced data items are passive. The distinction between active and passive data items is essential for the same reason that people referencing many other people may not be as famous as people that are being referenced by many other people.

Access: Semi-structured data spaces are commonly accessed through an initial full-text search followed by a brief exploration of directly and indirectly related data items. This data access method enables and encourages users to critically examine and validate search results through a more detailed investigation of near-matches. Semi-structured data spaces do not require the elimination of outdated data members whose often gratuitous continual growth consequently increases specific data access time. The data actuality is reflected through the changing network of relations. Data items with few or less significant relations to active network segments automatically become less accessible. Finally, semi-structured data spaces deal with the problem of data redundancy and preservation of authorship rights through individual data items being related rather than duplicated.

APPENDIX B THEORIES OF PSYCHOLOGY AND COGNITIVE SCIENCE

While the various observations about human perception and cognition in this work primarily draw from associationist, cognitivist and constructivist theories in psychology, the computational concepts are primarily portrayed from a connectionist point of view, which is a distinct area of research within the field of cognitive science. This appendix provides a brief overview of six different theories in psychology and introduces three distinct research approaches to cognitive science. This background information is helpful to more easily understand and situate the observations and concepts outlined in the section on Algorithms.

1. Theories of Psychology:

The various theories of psychology differ in their perspective on how human development and learning occurs. For example, psychologists portray differing perspectives about the influence of the physical and social environment on human development and learning. The interaction between biological and environmental factors is presented as a choice between "nature" and "nurture". Nature refers to the natural capacities of individuals while nurture refers to the physical and social environment. Many contemporary psychologies emphasize that nature and nurture must both be taken into account. Furthermore, psychologist have not yet found common ground on whether knowledge is given and absolute (subject to universal trends that all humans share) or constructed and relative (subject to individual, cultural, and ethnic variations). Also, psychologists differ in their views on whether human development and learning occurs in stages (gestalt and structuralist theories) or whether it is continuous (associationist theories).

1.1. Nativism:

The nativist view is that the structure of the mind cannot be explained but that the mind is a fixed entity that (like the body) does not gain significantly from experience. Nativists believe that the mind comes preconfigured with certain knowledge such as the concept of continuity (objects do not suddenly disappear in one place and reappear in another) or gravity (objects fall if left unsupported) [1]. In other words, nativists describe the development of knowledge as a process of "maturation" by which built-in capabilities unfold (or are unlocked) over time.

1.2. Associationism:

Unlike the Nativist, the associationists believe that knowledge comes from experience. The associationists further accept that humans are capable of reflecting on their experiences and construct associations. For example, to understand the concept of a bird, the human mind first discovers the similarities among multiple samples of birds and then manifests reoccurring observations by associating attributes such as "wings", "two legs", and "flying". Associationism is known for the concept of "law and effect" which means that a positive feedback leads to a repetition of an action while a negative feedback prevents the repetition of an action (consequently initiating a search for alternative actions). Unlike behaviorists, associationists assume that these evaluations not only take place based on external feedback but also through internal processes.

1.3. Behaviorism:

Behaviorism postulates an environmentally driven form of associationism. Behaviorists view knowledge as objective, given, and absolute [2] suggesting that only the minds input and output (behavior) rather than its inner mechanics can be analyzed. Instead, behaviorists are concerned with the observation of behavior and the adoption of organisms to the environment. Behaviorists do not demonstrate an interest in the discussion of mental processes and consequently consider learning "a passive and reactive process with learners responding to the expectations and requirements of the environment" [3]. For example, John Locke argued that children were born with "blank slates" (tabula rasa) and that adults would have complete control over their development [4].

Behaviorism dominated in the 1950's and 1960's. Early research in behaviorism mainly focused on stimulus-response experiments with animals. A famous experiment conducted by Ivan Petrovich Pavlov known as "classic conditioning" succeeded in making a dog salivate to the sound of a bell, linking the sound of the bell to the anticipation of food [5]. Burrhus Frederic Skinner, one of the leading behaviorists, experimented with what he called "operant conditioning" and "shaping behavior" [6]. "Unlike Pavlov's "classical conditioning", where an existing behavior (salivating for food) is shaped by associating it with a new stimulus (bell ringing), "operant conditioning" is the rewarding of only a partial behavior or a random act that approaches the desired behavior." [7] For example, if the goal is to make a pigeon turn to the left in a circle, a reward is given for any small movement to the left. Once the pigeon adapts to this concept, the reward is given for larger movements to the left until the pigeon accomplishes a full turn [7]. "Though most behaviorist experiments were conducted on the reflexive behavior of animals, behaviorist theories were generalized to higher-level functions of humans." [3] For example, Skinner tested his theories on how children learn to talk. He continued to reward a child for making a sound close to a desired word until the child would be able to say the word [7].

1.4. Cognitivism:

Cognitivists do not share the behaviorist's assumptions that learning can be reduced to outer-induced behavioral changes (stimulus-response model) but rather that learning takes place through active mental processing on the part of learners (information processing model). Cognitivists focus on modeling acts of thinking and learning by viewing knowledge as symbolic constructions [3] and propose a mental processing architecture similar to digital computers [8].

1.5. Constructivism:

"Whereas the behaviorists view knowledge as passive, primarily automatic responses to stimuli, and the cognitivists view knowledge as abstract symbolic representations, the constructivists view knowledge as a constructed entity developed by each individual learner [9]." [3] Whereas behaviorists and cognitivists view knowledge as given and absolute, constructivists consider knowledge to be relative, varying in time and space [3].

To constructivists, knowledge is internal and unique while information is external and interchangeable. Consequently, only information but not knowledge is transmitted among individuals. The conversion from knowledge into information as well as information into knowledge produces different results among individuals. Thus, the sender's knowledge differs from the receiver's knowledge. For example, if multiple people were to experience the same event they might remember it differently, meaning that even though the experience is the same the individual mental constructions of the experience differ. Essential to constructivist theories is the concept of stereotypes. For example, experiments with humans have revealed that when shown a picture of a male secretary many people remembered this person as the typewriter repair-man [1]. The two major theoretical approaches within constructivism originate from cognitive and social constructivism [10]. Cognitive constructivists argue that knowledge is a symbolic, mental representation in the mind of each individual while social constructivists consider the mind to be a distributed entity extending beyond the bounds of the human body into the social environment [3].

While Frederic Bartlett [11] pioneered what became the constructivist approach [12], the cognitive development theories introduced by Jean Piaget became the founding principles of the constructivist movement [3]. "Piaget's theory of cognitive development postulates that learning occurs through adaptation to interactions with the environment [13]." [3] Mental conflicts from the occurrence of new experiences cause a state of disequilibrium. This disequilibrium is resolved either through the addition of new experiences to existing bases of knowledge (assimilation) or through the modification of existing understanding (accommodation) [3]. Piaget outlined four levels of cognitive development [3]: 1. Sensorimotor: The intelligence in children from 0 to 2 years of age develops through movement and other sensory experiences. 2. Preoperational: The intelligence in children from 2 to 6 years of age develops through the use of pictures, words, and other symbols to represent objects and concepts. 3. Concrete operational: The cognitive development in children from 6 to 11 years of age includes logic as long as logic is applied to specific physical examples. 4. Formal operational: After the age of 11 the thinking of children matures to include the ability to understand and develop abstract concepts. At this stage, children (and adults) are capable of logical and abstract thinking without the need for physical examples on which to base their abstract ideas. Inspired by Piaget's work, researchers from various fields contributed their views and expertise to constructivism. For example, Nelson Goodman's [14] philosophical take on constructivism was that "contrary to common sense there is no unique 'real world' that preexists and is independent of human mental activity and human symbolic language" [15], but that "what we call the world is a product of some minds whose symbolic procedures construct the world" [15]. Marvin Minsky pointed out that all data-structures have their virtues and deficiencies and consequently postulated the importance of using multiple representations and parallel processing for the modeling of intelligent behavior [16]. Seymour Papert discovered that "some of the most crucial steps in mental growth are based not simply on acquiring new skills, but on acquiring new administrative ways to use what one already knows" [17].

1.6. Sociocognitivism:

Sociocognitivim is divided into socio-nativist, socio-associationist and socio-constructivist theories. Lev Vygotsky believed that "the mind does not primarily develop from the brain (nativism) or some general experiences (associationism) but through society [18]" [1]. For Vygotsky, the human mind advances through communication medias and techniques (thought sharing) such as number systems, spoken and written languages, social rules, and shared technologies [1]. Vygotsky's theory was that the acquisition of speech will fundamentally alter a human's attention, perception, thought and memory functions [1]. While Piaget viewed humans as individuals, Vygotsky considered humans as part of a group (society) [1]. The difference between these two views has been the subject of various experiments some of which concluded that children perform more successfully in groups than alone [1].

2. Theories of Cognitive Science:

Cognitive science complements research in psychology by taking advantage of emerging computer technologies capable of simulating and testing theories of the mind. Researchers in cognitive science believe that machines will eventually be able to process information and behave in ways that demonstrates intelligence. "Cognitive scientists have undoubtedly contributed to developmental theories by providing a wide range of running models, or simulations, of limited-yet-clearly-identifiable mechanisms of mind, which bring about novel insights as to how minds work" [19].

2.1. Symbolism:

Symbolism uses language-like symbolic processing as the means to understand, explain, and model human cognitive processes. Symbols are items that have a meaning or a function such as the words in this text. The symbols are combined according to specific rules such as the English language grammar. Even though common in classical AI, symbols are not restricted to words or numbers [20]. Some of the shortcomings of symbolism include [20]: 1. Symbolic representations are strictly propositional meaning they are difficult to use with non-language based applications such as applications for image processing. 2. Symbolic representations are not suitable for the modeling of concepts such as taste, sound, touch, and smell. 3. Symbolic representations cannot degrade meaning that a minor damage to a symbolic conceptual network may cause the loss of entire concepts. 4. Symbolic representations are not constructed based on statistical regularities found in the environment making them ineffective for the modeling of low-level perception. Pioneers in symbolic processing include Noam Chomsky [21] in Linguistics, Allen Newell and Herbert Simon [22] in Artificial Intelligence, Zenon Pylyshyn [23] in Psychology, and Jerry Fodor [24] in Philosophy [3].

2.2. Connectionism:

Connectionism hopes to explain the human mind with artificial neural networks. Artificial neural networks are composed of large numbers of nodes (analogs of neurons), links between nodes (analogs of synaptic connections between neurons through axons and dendrites), and adjustable weights that determine the strength of the links between the nodes (analogous to the effects of synapses between neurons). The two major approaches for the modeling of connectionist networks originate from localist and distributed representations [20]. Localist representations associate entire concepts (such as the concept of a bird) with one individual node in a network. Distributed representations on the other hand represent every concept through the interaction of multiple distributed (and decentralized) nodes across the network [25]. While distributed representations are neurologically more relevant than localist representations, they are often far more complex and difficult to analyze [20]. The functions (program) of a distributed representation are embodied in the configuration of the links [26]. Learning happens through changes to the strength of the links between the nodes (weights) [26]. Thus, the state of the strength of all links represents the long-term memory while the pattern of activity among the links represents the short-term memory [26]. Every change to a distributed representation affects the entire network. Consequently, a distributed representation is a continuously evolving system that never experiences the same state twice. Unlike symbolic networks, distributed representations only gradually adjust to change. This means that a change to a distributed representation can not trigger the loss of an entire concept but only increase or decrease the accuracy of a concept [25] making these distributed representations less vulnerable and more human-like.

During the 1940's, Walter McCullogh and Warren Pitts [27] introduced neural nets "as a way of understanding the nervous system". "These models were addressing how the brain works, rather than how the mind works". "Biologists and psychologists (proto-connectionists) began to apply these neural nets to higher cognitive tasks, such as learning, problem solving, and categorization" [28]. Donald Hebb's [29] research in neural biology is considered the beginning of cognitive psychobiology [3]. "During the 1950's and 1960's, Frank Rosenblatt [30] investigated the properties of mathematically described neural networks with modifiable connections, discovering the so-called 'perceptron convergence procedure' for training 'two-layer networks'" [31]. Rosenblatt demonstrated learning through machine recognition and identification of optical patterns [3]. In 1969, Minsky and Papert [32] showed that two-layer networks were limited in the functions they could compute, thereby providing a major decline in neural network research during the 1970's and 1980's [31]. In 1986, the introduction of the backpropagation algorithm for training "multi-layer networks" by David Rumelhart, Geoffrey Hinton, and Ronald Williams [33] laid the groundwork for a renaissance in neural net modeling [28]. This new concept allowed connectionist architectures to surpass classical symbolic approaches in various domains.

2.3. Dynamicism:

"Dynamicists believe that the brain is continually changing as it intersects with information from its environment [34]" [35] and that the mind functions operate independent of internal representations. Dynamicists explain cognition as a "multidimensional space of all possible thoughts and behaviors" [35]. Dynamicism applies concepts from dynamical systems theory to a description of cognition [35]. A dynamical system is a system that evolves over time and whose present state depends on rule-governed previous states [36]. "The evolution of the system, as a function of time and parameters, can be described in terms of differential functions or equations of motion" [36]. For example, dynamical systems theory could mathematically describe (and consequently predict) a stream of water as it finds its way through the topography of a terrain. The behavior of every water particle is influenced by its environment including other water particles. In turn, every water particle can also affect its environment. The dynamicists believe that the human brain operates and can be explained in the same way.

REFERENCES

SECTION A: EWALL

1 **NAVAIR** (Naval Air Systems Command). http://www.navair.navy.mil.
SPAWAR (Space and Naval Warfare Systems Center). http://www.spawar.navy.mil.
NPS (Naval Postgraduate School). http://www.nps.navy.mil.
NSW (Naval Special Warfare).
ARUP (http://www.arup.com).
Saab Aerospace (http://www.saabaerospace.com).

2 **CRS** (Caudill Rowlett and Scott) Center. Texas A&M University. http://crscenter.tamu.edu.

3 **Pena, William and Parshall, Steven. 2001.** Problem Seeking: An Architectural Programming Primer. John Wiley & Sons, 4th edition.

4 **Winnicott, D.W. 1982.** Playing and Reality. Routledge.

5 **DynaWall.** Frauenhofer Institute. http://www.darmstadt.gmd.de/ambiente/activities/dynawall.html (11/11/03).

6 **The Designers' Outpost.** Berkeley University. http://guir.berkeley.edu/projects/outpost (11/11/03).

7 **Interactive Workspaces Project.** Stanford University. http://iwork.stanford.edu (11/11/03).

8 **aire** (Agent-based Intelligent Reactive Environments Research Group at MIT). http://www.ai.mit.edu/projects/aire (11/11/03).

9 **Rekimoto, Jun and Saitoh, Masanori. 1999.** Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments. Proceedings of CHI'99.

10 **Inspiration.** http://www.inspiration.com (11/15/03).

11 **SmartDraw.** http://www.smartdraw.com (11/15/03).

12 **Visio.** http://www.microsoft.com/office/visio/prodinfo/default.mspx (11/15/03).

13 **Tinderbox.** http://www.eastgate.com/Tinderbox (11/15/03).

14 **OmniGraffle.** http://www.omnigroup.com/applications/omnigraffle (11/15/03).

15 **QuestMap.** http://www.compendiuminstitute.org/tools/questmap.htm (11/15/03).

16 **Greenberg, S. and Rounding, M. 2001.** The Notification Collage: Posting Information to Public and Personal Displays. Proceedings of the ACM Conference on Human Factors in Computing Systems, CHI Letters 3(1), 515-521, ACM Press.

17 **Keel, Paul and Huang, Jeffery. 2000.** NewsStripe: Digital News Board. Paper, unpublished.

18 **MeetingMiner.** IBM. http://www.research.ibm.com/thinkresearch/pages/2002/20020411_meetingminer.shtml (11/30/03).

19 **Fleming, Robert. 2003.** Information Exchange and Display in Asynchronous C2 Group Decision Making. SPAWAR (Space and Naval Warfare Systems Center), San Diego.

20 **Stasser, G. and Titus, W. 1985.** Pooling of Unshared Information in Group Decision Making: Biased Information Sampling During Group Discussion. Journal of Personality and Social Psychology, vol. 48, pp. 1467-1478.

21 **Gigone, Daniel and Hastie, Reid. 1993.** The Common Knowledge Effect: Information Sharing and Group Judgment. Journal of Personality and Social Psychology, vol. 65, pp. 959-974.

22 **Wittenbaum, M. / Hubbell, P. / Zuckerman, C. 1999.** Mutual Enhancement: Toward an Understanding of the Collective Preference for Shared Information. Journal of Personality and Social Psychology, vol. 77, pp. 967-978.
23 **Schrage, Michael. 1995.** No More Teams: Mastering the Dynamics of Creative Collaboration. Currency, Reprint edition.
24 **Lavery, T. A. / Franz, T. M. / Winquist, J. R. / Larson, J. R. 1999.** The role of information exchange in predicting group accuracy on a multiple judgment task. Basic and Applied Social Psychology, 21, 281-289.
25 **Reddy, Michael. 1979.** The Conduit Metaphor - A Case of Frame Conflict in Our Language about Language. Metaphor and Thought. Edited by Andrew Ortony. New York, NY: Cambridge University Press.
26 **Yee, Susan and Keel, Paul. 1995.** Inventing Organizations of the 21th Century: Design Communication in Architectural Practice. Unpublished.
27 **PACOM** (U.S. Pacific Command). http://www.pacom.mil (01/02/04).

SECTION B: ALGORITHMS

1 **Quan, Dennis / Huynh, David / Sinha, Vineet / Karger, David / Zhurakhinskaya, Marina. 2002.** Basic Concepts for Managing Semi-structured Information in Haystack. Cambridge, MA: MIT, Student Oxygen Workshop.
2 **Lee, Jintai. 1990.** SIBYL: A Tool for Managing Group Design Rationale. CSCW '90, Proceedings of the Conference on Computer Supported Cooperative Work. Los Angeles, CA: ACM. pp. 79-92.
3 **Minsky, Marvin. 1988.** The Society of Mind. Touchstone Books.
4 **Visio.** http://www.microsoft.com/office/visio (02/24/04).
5 **SmartDraw.** http://www.smartdraw.com (02/24/04).
6 **Inspiration.** http://www.inspiration.com (02/24/04).
7 **Baumgartner, Jason / Zou, Yuezheng / Börner, Katy. 2002.** Spring Embedding Algorithm. http://ella.slis.indiana.edu/~katy/L697/code/spring.html (02/24/04).
8 **Graph Layout Toolkit.** http://www.tomsawyer.com/glt (02/24/04).
9 **Keel, Paul. 1996.** Process and relation analysis: Capturing architectural thought. Thesis, MIT.
10 **Indurkhaya, Bipin. 1992.** Metaphor and cognition. Dordrecht, Kluwer Academic Publishers.
11 **Nueckles, Matthias and Janetzko, Dietmar. 1997.** Shifting from Analysis to Synthesis: Evidence for Cascaded Interpretations in Metaphor Comprehension. In Proceedings of the 2nd European Conference on Cognitive Science, pp. 14-19. Manchester, U. K.
12 **Pena, William and Parshall, Steven. 2001.** Problem Seeking: An Architectural Programming Primer. John Wiley & Sons, 4th edition.
13 **Wertheimer, Max. 1978.** Productive Thinking. Greenwood Publishing Group, Reprint edition.
14 **Wilson, Robert and Keil, Frank (Eds.). 1999.** The MIT Encyclopedia of the Cognitive Sciences. Cambridge, MA: MIT Press.
15 **Kirsh, David. 1995.** Complementary Strategies: Why we use our hands when we think. In Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society. Hillsdale, NJ: Lawrence Erlbaum.

16 **Wertheimer, Max. 2000.** Laws of Organization in Perceptual Forms. In Visual Perception: Essential Readings (Key Readings in Cognition). Yantis, Steven (Ed.). Philadelphia: Psychology Press, pp. 216-224.
17 **Dhande, Sheel. 2002.** Thesis Proposal. MIT.
18 **Zobrist, Albert and Thompson, William. 1975.** Building a distance function for Gestalt grouping. IEEE Transactions on Computers, vol. C-24, no. 7, pp. 718–728.
19 **Lowe, David. 1985.** Perceptual Organization and Visual Recognition. Kluwer Academic Publishers.
20 **Mohan, Rakesh and Nevatia, Ramakant. 1992.** Perceptual Organization for Scene Segmentation and Description. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 6, pp. 616-635.
21 **Amir, Arnon and Lindenbaum, Michael. 1998.** A generic grouping algorithm and its quantitative analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 2, pp. 168-185.
22 **Saund, Eric / Mahoney, James / Fleet, David / Larner, Dan / Lank, Edward. 2001.** Perceptual Organization as a Foundation for Graphics Recognition. Fourth IAPR International Workshop on Graphics Recognition. Kingston, Ontario.
23 **Ullman, Shimon. 1966.** Visual Cognition and Visual Routines. In High Level Vision. MIT Press, chap. 9, pp. 263-315.
24 **Johnson, Mark. 2001.** Functional Brain Development in Humans. Nature Reviews / Neuroscience, vol. 2, July 2001, pp. 475-483.
25 **Gruber, Howard and VonEche, Jacques (Eds.). 1977.** The essential Piaget. Basic Books, Inc. pp. 353-358, 299-301.
26 **Frith, Uta. 1998.** Literally changing the brain. Brain, vol. 121, pp. 1011-1012.
27 **Schoen, Donald. 1983.** The reflectiove practitioner: How professionals think in action. Basic Books, Inc. pp. 309-315.
28 **Zweig, Stefan. 2000.** The Royal Game. Holmes & Meier Publishers, Inc., Reprint edition.
29 **Brainwashed** (Schachnovelle). 1960. Tapeworm.
30 **HTML** (Hyper Text Markup Language). http://www.w3.org/MarkUp (02/24/04).
31 **XML** (Extensible Markup Language). http://www.w3.org/XML (02/24/04).
32 **Keel Paul. 2000.** DataBox: Unstructured Data Space. Paper, unpublished.
33 **Levy, Pierre and Bononno, Robert. 1997.** Collective Intelligence. Perseus Publishing.
34 **Google** Technology Overview. http://www.google.com/press/overview_tech.html (10/05/03).
35 **Amazon.** http://www.amazon.com (02/24/04).
36 **Freud, Sigmund. 1900.** The Interpretation of Dreams.
37 **Edelson, Edward. 1992.** Sleep. New York, Chelsea House. pp. 14, 48-51, 63.
38 **Hobson, Allan and McCarley, Robert. 1977.** The brain as a dream state generator: An activation-synthesis hypothesis of the dream process. The American Journal of Phychiatry, vol. 134, pp. 1335-1348.
39 **Sutterlin, Emilie et al. 1999.** Sleep from A to Zzz. http://library.thinkquest.org/25553/english/dream/psycho/index.shtml?tqskip1=1&tqtime=0509 (05/09/03).
40 **Crick, Francis and Mitchison, Graeme. 1983.** The function of dream sleep. Nature, 304 (14), July, pp. 111-114.
41 **Winson, Jonathan. 1985.** Brain and Psyche: The Biology of the Unconscious. Anchor Press.

42 **Bachelard, Gaston. 1960.** La Poetiqué de la Rêverie. Paris: Presses Universitairs de France. Translation by D. Russell, New York: Orion Press.
43 **Geertz, Clifford. 1973.** Thick description: Towards an Interpretive Theory of Culture. The Interpretation of Cultures. Basic Books: New York, pp. 3-30.
44 **Uzuner, Ozlem / Davis, Randall / Katz, Boris. 2002.** Recognizing Text Similarity. http://www.ai.mit.edu/research/abstracts/abstracts2002/natural-language/11uzuner.pdf (07/05/03).
45 **Minsky, Marvin. 1985.** The society of mind. NY: Simon & Schuster. p. 64.
46 **Bevilacqua, John. 2004.** Decision Making by Agent Committee. Master Thesis, MIT.
47 **McGraw, K.M. and Steenbergen, M. 1995.** Pictures in the head: Memory representations of political candidates. In M. Lodge and K.M. McGraw (Eds.) Political judgment: Structure and process, pp. 15-41. Ann Arbor: University of Michigan Press.
48 **Abiteboul, Serge. 1997.** Querying Semi-Structured Data. ICDT 1997, pp. 1-18.
49 **Cluet, Sophie. 1997.** Modeling and Querying Semi-structured Data. SCIE 1997, pp. 192-213.
50 **QBIC: IBM's Query By Image Content.** http://wwwqbic.almaden.ibm.com (03/02/04).
51 **Ullman, Shimon. 1995.** Sequence Seeking and Counter Streams: A Model for Information Flow in the Visual Cortex. Cerebral Cortex V.5, February 1995, pp. 1-10.
52 **Sack, Warren. 2000.** Design for Very Large-Scale Conversations. Ph.D. Dissertation. Cambridge, MA: MIT Media Laboratory.
53 **Wenz, Florian and Hirschberg, Urs.** Phase(x) - Memetic Engineering for Architecture. http://caad.arch.ethz.ch/~Ehirsch/ecaade (02/26/04).
54 **inxight Star Tree.** http://www.inxight.com/map (03/23/04).
55 **The Brain.** http://www.thebrain.com (03/23/04).
56 **WebMap.** http://www.webmap.com (03/23/04).
57 **Kart00.** http://www.kartoo.net/e/en/produits.php?d=sitebox&p=intro (03/23/04).
58 **Donath, Judith. 1995.** Visual Who: Animating the affinities and activities of an electronic community. ACM Multimedia 95. California: San Francisco: November 5-9.
59 **Killian, Axel. 2000.** Defining Digital Space Through a Visual Language. MIT, SMArchS Thesis. http://destech.mit.edu/akilian/projectpages/thesis2000.html (04/06/04).
60 **Graph Layout Toolkit for the C++ Development Community.** http://www.tomsawyer.com/glt (04/06/04).
61 **Baumgartner, Jason / Zou, Yuezheng / Börner, Katy. 2002.** Spring Embedding Algorithm. http://ella.slis.indiana.edu/~Ekaty/L697/code/spring.html (04/06/04).
62 **Kohonen, Teuvo et al. 1999.** WEBSOM: Self-Organizing Maps for Internet Exploration. http://websom.hut.fi/websom (04/06/04).
63 **SmartMoney.** http://www.smartmoney.com/maps (04/06/04).
64 **map.net.** http://maps.map.net (04/06/04).
65 **ThemeScape.** http://www.cartia.com.
66 **City'O'Scope.** http://www.macrofocus.com/cityoscope/index.html (04/06/04).

APPENDIX B: THEORIES OF PSYCHOLOGY AND COGNITIVE SCIENCE

1 **Mann, John. Book Review: Richardson, Ken. 1998.** Models Of Cognitive Development. Psychology Press. http://www.fountain.btinternet.co.uk/stmarys/reviews/models.html (05/07/03).

2 **Graham, George. 2002.** Behaviorism. The Stanford Encyclopedia of Philosophy, Fall 2002 Edition. Zalta, Edward N. (Ed.), http://plato.stanford.edu/archives/fall2002/entries/behaviorism (05/04/03).

3 **Dara-Abrams, Benay Phyllis. 2002.** Theoretical Foundation in Educational Psychology for Multi-Intelligent Online Learning. http://www.brainjolt.com/docs/foundation.pdf (04/29/03).

4 **Wolterstorff, Nicholas. 1996.** John Locke and the Ethics of Belief. Cambridge University Press.

5 **Tiner John, and Babkin, Boris. 1975.** Pavlov: A Biography. Toronto, Canada: The University of Chicago Press.

6 **Skinner, Burrhus Frederic. 1985.** Particulars of My Life, the Shaping of a Behaviorist, a Matter of Consequences. New York University Press. p. 56.

7 A Science Odyssey. People and Discoveries. B.F. Skinner. http://www.pbs.org/wgbh/aso/databank/entries/bhskin.html (05/07/03).

8 **Gardner, Howard. 1987.** The Mind's New Science: A History of the Cognitive Revolution. NY: Basic Books.

9 **Houghton Mifflin Company. 2001.** Constructivism. Teacher Education Station. http://college.hmco.com/education/station/concept/construct/conback.html (02/23/01).

10 **Chen, Irene. 2000.** An Electronic Textbook on Instructional Technology. University of Houston. College of Education. http://www.coe.uh.edu/~ichen/ebook/ET-IT/cover.htm (02/23/01).

11 **Bartlett, Frederic Charles. 1932.** Remembering: An experimental and social study. Cambridge: Cambridge University Press.

12 **Good, Thomas and Brophy, Jere. 1990.** Educational psychology: A realistic approach. NY: Longman.

13 **Piaget, Jean and Inhelder, Barbel. 1969.** The Psychology of the Child. NY: Basic Books.

14 **Goodman, Norman. 1978.** Ways of Worldmaking. Indianapolis, IN: Hackett.

15 **Bruner, Jerome. 1986.** Actual Minds, Possible Worlds. Cambridge, MA: Harvard University Press. p. 95.

16 **Minsky, Marvin. 1990.** Logical vs. Analogical or Symbolic vs. Connectionist or Neat vs. Scruffy. In Winston, Patrick (Ed.), Artificial Intelligence at MIT: Expanding Frontiers, 1. Cambridge, MA: MIT Press. p. 4.

17 **Minsky, Marvin. 1985.** The society of mind. NY: Simon & Schuster. p. 102.

18 **Vygotsky, Lev. 1978.** Mind in Society: The Development of Higher Psychological Processes. Cambridge, MA: Harvard University Press.

19 **Ackermann, Edith. 1998.** New trends in Cognitive Development: Theoretical and Empirical Contributions. In Demetriou, A. (Ed.), Cognitive development: Steps toward developmental cognitive science. Special Issue of Learning and Instruction: The Journal of the European Association for Research on Learning and Instruction. vol.8, 4, pp. 375-385.

20 **Eliasmith, Chris (Ed.). 2002.** Dictionary of Philosophy of Mind. http://www.artsci.wustl.edu/~philos/MindDict (04/29/03).

21 **Chomsky, Noam. 1957.** Syntactic Structures. The Hague: Mouton.
22 **Newell, Allen and Simon, Herman. 1972.** Human Problem Solving. Englewood Cliffs, NJ: Prentice-Hall.
23 **Pylyshyn, Zenon. 1973.** What the mind's eye tells the mind's brain. Psychological Bulletin, vol. 80. pp. 1-24.
24 **Fodor, Jerry. 1975.** The Language of Thought. Cambridge, MA: Harvard University Press.
25 **Churchland, Patricia Smith and Sejnowski, Terrence. 1992.** The computational brain. Cambridge, MA: MIT Press.
26 **Meeden, Lisa.** Introduction to Connectionism. http://yoda.cis.temple.edu:8080/UGAIWWW/lectures95/meeden/nnet/subsection3_1_1.html (05/10/03).
27 **McCulloch, Walter and Pitts, Warren. 1943.** A Logical Calculus of Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics, vol. 5, pp. 115-133. Chicago: University of Chicago Press.
28 **Chaput, Harold Henry. 1998.** Modeling Development with a Constructivist Learning Architecture. http://www.cs.utexas.edu/users/chaput/research/PA/welcome.html (05/11/03).
29 **Hebb, Donald. 1949.** The organization of behavior; a neuropsychological theory. NY: Wiley-Interscience.
30 **Rosenblatt, Frank. 1958.** The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychological Review, vol. 65, no. 6, pp. 386-408.
31 **Aizawa, Ken. 2002.** Dictionary of Philosophy of Mind. http://www.artsci.wustl.edu/~philos/MindDict (04/29/03).
32 **Minsky, Marvin and Papert, Seymour. 1969.** Perceptrons. Cambridge, MA: MIT Press.
33 **Rumelhart, David / Hinton, Geoffrey / Williams Ronald. 1986.** Learning representations by back- propagating errors. Nature, vol. 323, pp. 533-536.
34 **van Gelder, Timothy and Port, Robert. 1995.** It's about time: Overview of the dynamical approach to cognition. In Port, Robert and van Gelder, Timothy (Eds.), Mind as Motion: Explorations in the Dynamics of Cognition. Bradford Books, MIT Press.
35 **Eliasmith, Chris. 1995.** Mind as a Dynamical System. Thesis. Ontario, Canada: University of Waterloo. p. 2.
36 **van Rooij, Iris. 2003.** Dynamical Systems Theory & Gestalt Perception. http://web.uvic.ca/~irisvr/DynamicGestalt.htm (05/23/03).

RELATED WORK

1 **Kahan, Michael. 2004.** Computer Analysis, Learning and Creation of Arrangements of Information. Master Thesis, MIT.
2 **Li, Yao. 2004.** Discovering Structure of Data to Create Multiple Perspective Visualization. Master Thesis, MIT.
3 **Bevilacqua, John. 2004.** Decision Making by Agent Committee. Master Thesis, MIT.
4 **Dalsass, Charles. 1996.** Applying Information Technologies to Architectural User Programming. Master Thesis, MIT.
5 **Keel, Paul. 1997.** Process and relation analysis: Capturing architectural thought. Master Thesis, MIT.