




~ EXPERIENTIAL COMPUTING [By Ramesh Jain]

SEEKING INSIGHT FROM DATA, USERS EXPERIENCE, EXPLORE, AND
EXPERIMENT, NO LONGER LIMITED TO JUST GENERATING LISTS OF
POSSIBLE ANSWERS TO SIMPLISTIC QUERIES.



SOLUTIONS ARE NEEDED FOR THE TECHNOLOGICAL challenges that are paradoxically the result of the recent tremendous progress in information and communication technology (ICT), sensors, and embedded computing. Computing began with a focus on data and later shifted to information and communication. To address the requirements of today's emerging network-based applications, the focus must shift again, this time to experience and insight.

My research group at the Georgia Institute of Technology is thus developing a body of computing approaches called experiential computing. Decision makers at all organizational levels routinely using computers need insights that can come only from their own experience and experimentation with all available data sources. They must be able to explore and experience events from multiple perspectives and revisit them as often as needed to obtain that insight. In an experiential computing environment, users apply their senses directly, observing event-related data and information of interest. Moreover, users explore the data by following their own personal interests within the context of an event.

Experiential environments free users from the tedium of managing enormous volumes of disparate heterogeneous data. They don't try to interpret an experience; instead, they provide an environment that can be used to naturally understand events, somewhat like computer-aided design systems provide designers environments that are supportive of their design efforts.

Most computing techniques today, including supercomputers and the Web, fundamentally result

from such driving applications as scientific computations, business decision making, network computing, and personal computing. The experiential applications I describe here characterize the demand increasingly placed on emerging computational techniques and infrastructure, including distributed middleware and the grid.

Situation monitoring and data warehouses. Business-activity monitoring [5, 6], bioinformatics [10], homeland security, and other such applications draw from a vast network of disparate data sources, including databases, sensors, and systems in which data is entered manually. All produce streams of data for a variety of applications; for example, seismologists might use them to help warn of impending earthquakes, and businesspeople might use them to monitor the status of activities at dispersed locations or to analyze the causes of past market events. Real-time data analysis must be combined with real-time data assimilation from all sources in order to present a unified model of the situation in some intuitive form. Techniques and tools developed for payroll databases are, for example, inadequate for this environment. Data mining techniques are suitable after a hypothesis has been formed, but visualization and interactive tools must be used first to help generate the hypothesis.

Personalized event experience. Most storytelling, whether for educational or entertainment purposes, presents an experience from the storyteller's perspective. The user's experience is secondary and bound by the storyteller's perspective. But users should be able to explore and experience events from multiple perspectives and revisit these events

A HUMAN PARTICIPANT PARTS VIRTUAL CURTAINS TO LOOK OUT A WINDOW INSIDE A VIRTUAL ENVIRONMENT. THE 3D VIRTUAL REPRESENTATION WAS GENERATED IN REAL TIME FROM MULTIPLE CAMERAS; THE RESULTING AVATAR IS CAPABLE OF NATURALLY INTERACTING WITH VIRTUAL OBJECTS AND SIMULATIONS. NOTE THAT SINCE THE AVATAR IS A SHAPE APPROXIMATION (VISUAL HULL), SHAPE ERRORS—IMAGE NOISE, JAGGIES, AND DARK AREAS—APPEAR IN THE RECONSTRUCTION. (IMAGE CREATED BY BENJAMIN LOK, SAMIR NAIK, MARY WHITTON, AND FREDERICK P. BROOKS, JR. IN THE EFFECTIVE VIRTUAL ENVIRONMENTS GROUP AT THE UNIVERSITY OF NORTH CAROLINA AT CHAPEL HILL, 2002.)

as many times as they wish to obtain the desired insight. Users who want to understand from their own perspective how an event evolved would benefit from viewing, say, a quick animation of statistics related to several categories or a particular event from multiple perspectives. They might also listen to descriptions of the same event by multiple commentators. Each of these operations represents a query the event record must answer by presenting the results in a form users find familiar and immediately usable.

EventWeb. The Web today consists of pages prepared predominantly in document mode, but language is a significant barrier in a text-based Web. A

IN an experiential environment, users apply their natural human senses directly to observe data and information of interest related to a particular event.

Web of events would be universal because users would experience the events in the medium—text, video, audio, or some combination—they find most appealing and useful at the moment. In such a Web, each node represents an event, whether past, current, or future. Users might post events on this Web by connecting one or more cameras, microphones, infrared and other sensors, databases, and related textual information to let visitors experience them as they wish. For each node, all information from the various sources would be unified and presented independent of the sources. An EventWeb would be independent of language and have much greater appeal among the 90% of the world's population that lacks access to current ICT due to language and educational barriers.

Folk computing. The income disparity between the richest and poorest countries is increasing and is also at the root of the digital divide—an economic and social gap not only unfortunate for the have-nots but potentially dangerous for everyone else [3]. Information and timely communication are essential for the proper use and distribution of scarce resources. Technology is and always has been the vehicle for com-

munication; countries without it find themselves on the dark side of the digital divide. To bridge the gap, researchers must find ways to package ICT so it reaches everyone, even the illiterate and the poor. Folk computing for everyone reflects a deep understanding of individual needs, using technology relevant to local living conditions, socioeconomic status, education, and language. It is not a matter of recycling products that work in the developed economies.

ICT products, including the Web, depend on their users' ability to read and write some language, usually English. But many people in developing countries are illiterate even in the languages they speak. A more realistic solution for them would be information and communication devices that use audio, video, and tactile input and output and let their users work in their natural environments.

These emerging applications include and reflect the following data characteristics and operational trends:

- Spatiotemporal and live data streams are the norm rather than the exception;
- Examining a holistic picture of a situation is more productive than examining silos of related data;
- Users seek insights and information independent of media and data sources;
- Users do not want information that is not immediately relevant to their particular interests; and
- Exploration, not querying, is the predominant mode of interaction.

Exploiting Human-Machine Synergy

Most conventional information environments actually work against human-machine synergy [7]. The human mind is very efficient at conceptual and perceptual analysis and relatively weak at mathematical and logical analysis; computers are the opposite. The process of designing information environments involves logical and mathematical principles; humans eventually interact with these systems using a logical approach. Even the interfaces are designed to expect users to formulate anything but the most obvious searches using logical combinations, thus discouraging many users.

The nature and volume of the data our computing systems deal with never stop changing due to the demands of emerging applications. Depending on the volume of data, what we do with the data itself also changes. The evolving nature of data sources and users' desired operations are captured in Table 1. Data used to come from a clearly identified source, but the Web has completely changed data access and user

expectations. Similarly, when data volumes are small, people are interested either in data or information; when data volumes are large and diverse, they want insights. This relationship between users and data has profound implications for ICT; for example, if databases are effective for getting precise information from a single data source, why would anyone keep using them to try to derive insights from multiple sources? Visualization environments and interactive tools are useful for gaining insights from specific identifiable sources, but most of today's emerging applications fall in the top right quadrant of the table. To help generate insights from multiple heterogeneous sources, an experiential environment has to unite disparate data sources and free decision makers to explore their own perceptions. In experiential environments, users apply their senses directly to observe data and information of interest related to a particular event; they interact naturally with the data based on their own personal interests in the context of the event.

There is a very clear trend in the evolution of computing approaches from databases to search engines. The fundamental differences between these systems were shown in [1], which compared their key characteristics. Experiential environments extend the trend to now produce those longed-for insights, as outlined in Table 2. Reflecting humans' perceptual and cognitive strengths, experiential environments involve several key characteristics:

They are direct. An experiential environment provides a holistic view of an event without using arcane metaphors and commands. Users are in a familiar environment in which they use natural actions based on familiar operations and their anticipated results. Data is readily interpreted by users' human senses; users then interact with the dataset to produce modified datasets.

They provide the same query and presentation spaces. Most current information systems employ different query and presentation spaces. For example, popular search engines provide query windows for users to enter

keywords; they then respond with lists of perhaps thousands of entries on hundreds of pages. Users have no idea how the entries on the first page are related to the entries on the 16th page, how many times the same entry appears, or even how the entries on the same page are related to one another. With spreadsheets, users articulate queries by changing certain data displayed in the context of other data items. Their actions result in an updated spreadsheet showing new relationships. Query and presentation spaces are the same in these systems—what-you-see-is-what-you-get, or WYSIWYG.

They consider both user state and query context. Ideally any system should know the state and context of its users and present user-relevant information in the given state and context. People operate best in known contexts and tend to lose focus when confronted by context switching. Information systems, including databases, are designed to provide scalability and efficiency. These considerations have led to a legacy of stateless systems, that is, they have no memory, and each query is a new query. Statelessness is the reason users often cite for dissatisfaction with today's generation of search engines; they ignore the state of the user.

They promote perceptual analysis and exploration. Because users employ their senses to analyze, explore, and interact, the system becomes more compelling and understandable. Text-based systems provide abstract information in visual form. Video games and many simulation systems are engaging because they provide users with powerful visual environments, along with sound and in some cases tactile input.

Research Challenges

Because experiential systems are so different from their conventional counterpart data-access systems, their designers must rethink nearly every one of their components (see Figure 1), including:

Data acquisition and analysis. Experiential environments draw data and information from many disparate sources, ranging from text to sensory input. Their designers must therefore emphasize the semantic and contextual processing of heterogeneous data. They also require effective middleware for adding, deleting, and modifying disparate data sources and for processing these sources for specific operations.

Assimilation. Sophisticated control and communi-

Insight	Visualization	Experiential Environments
	Current Databases	Search Engines
Information	Precise Sources (20 th Century)	Imprecise Sources (21 st Century)

Table 1. Evolving nature of data sources and users' desired outcomes.

	Database	Information Retrieval	Experiential Environment
User's Query	Specific	General	Emergent
System Provides	Data item	Pointer	Heterogeneous data
Retrieval Method	Deterministic	Probabilistic	Hybrid
Success Criterion	Efficiency	Utility	User Satisfaction

Table 2. Data transformed into information and now transformed into insight.

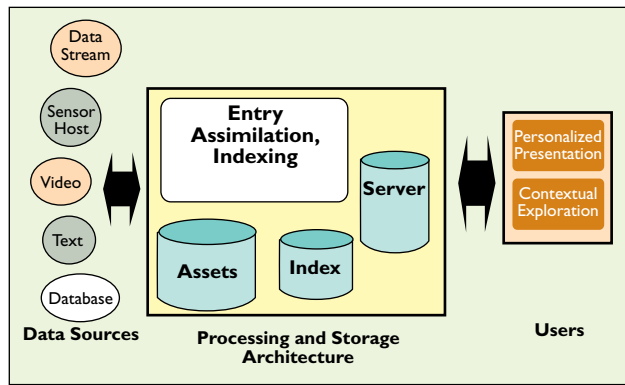


Figure 1. Architecture of an experiential computing system.

ation systems assimilate data from disparate sources using domain-model-based techniques (such as Kalman filtering) [8]. The mathematical domain model is at the heart of these systems. Development of model-based techniques that can be generalized via powerful domain modeling tools are essential for building next-generation systems capable of employing live heterogeneous data. The domain models described here are more complex and must deal with the so-called signal-to-symbol gap, that is, with data ranging from analog signals to highly abstracted symbols.

Unified indexing. Conventional indexing techniques result in data silos based on types of data. What is required for developing experiential systems is a unified indexing approach capable of indexing disparate data sources based on domain semantics rather than on media type [11, 12]. These techniques allow representation of heterogeneous knowledge by using domain semantics. This representation technique facilitates management of tacit knowledge, making possible many new applications of knowledge management approaches.

Exploration environment. Experiential environments can't be developed as simple interface mechanisms. Conventional query-based environments (such as search engine interfaces) have been useful with databases and search-oriented applications, including the Web. However, experiential applications require holistic pictures and exploratory environments. Query and presentation spaces must function in the same way, resulting in WYSIWYG environments for search and exploration. Incorporating user context (including location and time) and sequences of operations is essential for making these environments natural for users to interact with through their human senses. This development philosophy requires the exploration environment to maintain the state of the interaction; conventional systems have always been designed to be stateless for performance and scalability reasons.

Personalized presentations. If we want computer technology to cross the chasm and become usable by the world's vast populations, then data engineers must design these environments. Personalized presentations and exploratory environments require consideration not only of user interfaces but of media synchronization, media summarization, and other complicated engineering issues closely related to data organization and processing. Most personalization systems today focus on users' static characteristics. User context plays an equally important role in a system's approach to determining which information using which media types in which sequence should be presented to which user.

Early Experiential Systems

My colleagues and I have developed an event-oriented approach for implementing experiential systems using event-based domain models to construct a new index independent of the data types in different data sources [9]. An event is defined as a significant occurrence or happening at a single point in space-time. An application domain can be modeled in terms of events and objects. Events are hierarchical and include all the desirable characteristics that have made objects so popular in software development. In fact, events can be viewed as objects with time and space as their primary attributes.

A unified index is needed to organize multifarious data related to an event; we call such an index an eventbase. The eventbase contains all the information

EVENTS can be viewed as objects with time and space as their primary attributes.

that has been assimilated from various sources linking to the original data sources. The links are especially important for presenting appropriate media in the context of events. Having users interact directly with the eventbase has several advantages, including: (a) preprocessing important information related to events and objects based on domain knowledge; (b) presenting information using domain-based visualization; and (c) providing unified access to all information related to an event independent of the time the data became available or was entered into an associated database. As discussed in [2], these performance

characteristics allow users of experiential environments to access information.

My colleagues and I at Praja, Inc., developed EventViewer in 1999 to provide a WYSIWYG environment for experimenting with these ideas; Figure 2 is an EventViewer application screen. EventViewer offers multidimensional navigational and exploration capability in a WYSIWYG environment. Each event includes three basic characteristics: name and class; the location where it took place; and the time it took place.

Users navigate through the class ontology hierarchies. Navigation through location and time employs either zooming or moving in different directions, including left, right, up, and down. Users can select parts of maps ranging from the corner of a room to the whole world. Similarly, on the timeline, users can select microseconds, centuries, even light-years. Once a user selects the event classes, a map in the location space, and time on the timeline, the system displays all events and their selected attributes at three places in three formats: a list in the space provided for event list; symbols on the location map; and symbols at the appropriate times on the timeline. These displays are tightly linked. Thus, if a user selects an item in the event list, it will be selected and highlighted in a different color; the location and time spaces follow the same selection process.

Displaying events on both a map and a timeline maintains their contexts for users who might choose to refine the search criteria, thus yielding more refined results. This criteria/results refinement feature allows users to experiment with the dataset, prompting insights and helping them form hypotheses. It might also be linked with data mining tools to explore large data warehouses. Users who want to know more about particular events can explore them by double-clicking on the ones listed in any of the three display areas. The user is automatically presented with event details, along with all data sources (such as audio,

video, and text) and other characteristics.

Two applications developed at Praja, Inc., from 1999 to 2002 give an idea of what might be done in this environment:

Demand activity monitoring. In a modern enterprise, line managers need to quickly and dependably identify potential problem areas, how they developed, and how things should be changed in the future. The



Figure 2. Screenshot of an EventViewer for demand activity monitoring providing a WYSIWYG search and exploration environment.

focus is on “performance indicators,” or the discrepancies between planned and actual performance, and the interrelationships among them and the available infrastructure, as well as environmental factors and promotional efforts. Normally, just finding potential problems is insufficient. The context in which the problem occurred is important, too. Context includes related activities, as well as historical perspective; key activities are sales and inventory (monthly, daily, and hourly for various geographic regions). Figure 2 is a screenshot of EventViewer for this application. Performance indicators for each activity are mapped to red, yellow, and green, based on domain-specific criteria.

Football highlights. This application gave football fans of 25 U.S. universities an environment for exploring and locating the most interesting parts of a particular game. A game can be modeled as an event-graph with several levels of event hierarchies and transitions between events determined by what happens in the game. We used video (and audio) from multiple cameras located in stadiums in which the games were being played, play-by-play information gener-



Figure 3. Experiential environment for football fans in a time-machine format.

ated by a number of companies, including Stats, Inc., as a data stream, and access to player and statistics databases.

The system parsed the play-by-play data stream, applied the rule base to it, and presented to users an eventbase of the game as the time machine in Figure 3. Users could go to any moment in time and view all the statistics and other relevant game data. They could filter events and view them in standard football representations. Double-clicking on a particular play would, for example, produce more information, including video of the play. Thus, users could view scoring plays resulting in touchdowns by their favorite teams and videos of their favorite plays from multiple angles.

Conclusion

Experiential computing represents a natural step in assimilating, understanding, and using the flood of data available to us all. Employing new techniques to deal with live spatiotemporal data streams will enable ICT to address many real-world problems, including homeland security and real-time enterprise monitoring. On the other hand, experiential environments will also bring computing to billions of underprivileged, even illiterate, people worldwide by providing natural relatively language-independent interfaces to computing devices.

Applying ICT to problems in mainstream human society across all socioeconomic strata also produces

new scientific and business challenges. Research is being done in all these areas, but like the well-known fable of the elephant and the six blind men, a holistic perspective is essential for addressing them. Techniques developed in one element of a system that ignores other elements are likely to fail or provide only limited applicability. Similarly, developing interfaces for experiential environments without also developing and providing data assimilation, indexing, and management techniques will not be scalable or efficient [1]. **C**

REFERENCES

1. Belew, R. *Finding Out About: A Cognitive Perspective on Search Technology and the WWW*. Cambridge University Press, Cambridge, U.K., 2000.
2. Jain, R. Events in heterogeneous data environments. In *Proceedings of the International Conference on Data Engineering* (Bangalore, India, Mar.). IEEE Computer Society Press, Los Alamitos, CA, 2003, 8–21.
3. Jain, R. Folk computing. *Commun. ACM* 46, 4 (Apr. 2003), 27–29.
4. Katkere, A., Schlenzig, J., Gupta, A., and Jain, R. Interactive video on WWW: Beyond VCR-like interfaces. *Comput. Networks and ISDN Syst.* 28 (1996), 1559–1572.
5. Luckham, D. *Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley, Reading, MA, 2002.
6. McCoy, D. and Govekar, M. *Evolving Interaction Styles in Business Activity Monitoring*. Gartner Group, Stamford, CT, Mar. 2002.
7. Norman, D. *The Invisible Computer*. MIT Press, Cambridge, MA, 1999.
8. Roth, Y. and Jain, R. Knowledge caching for sensor-based systems. *Artif. Intelli.* (Jan. 1994), 2–24.
9. Santini, S. and Jain, R. Semantic interactivity in presence systems. In *Proceedings of the 3rd International Workshop on Cooperative and Distributed Vision* (Kyoto, Japan, Nov. 1999).
10. Singh, R. An overview of computational knowledge discovery and pattern analysis problems in contemporary drug discovery and design. In *Proceedings of the DIMACS Summer School Tutorial on New Frontiers in Data Mining* (Rutgers University, Piscataway, NJ, Aug. 2001).
11. Sonnen, D. and Morris, H. *Addressing Requirements for Unified Data Access*. International Data Corp., Framingham, MA, Aug. 2001.
12. Sonnen, D. and Morris, H. *A Unified View of Information Across the Enterprise: Moving Beyond the Data Integration Paradigm*. International Data Corp., Framingham, MA, Jan. 2001.

RAMESH JAIN (jain@ece.gatech.edu) is the Farmer Distinguished Chair and GRA Eminent Scholar in the School of Electrical and Computer Engineering at the Georgia Institute of Technology, Atlanta.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.