State-of-the-Art Survey

LNAI 3345

Yang Cai (Ed.)

Ambient Intelligence for Scientific Discovery

Foundations, Theories, and Systems

Springer

Yang Cai (Ed.)

Ambient Intelligence for Scientific Discovery

Foundations, Theories, and Systems

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editor

Yang Cai
Carnegie Mellon University, School of Computer Science
5000 Forbes Avenue, Pittsburgh, PA 15213, USA
E-mail: ycai@cmu.edu

Library of Congress Control Number: 2004117781

CR Subject Classification (1998): I.2, H.2.8, H.4, H.5, I.3, C.2

ISSN 0302-9743
ISBN 3-540-24466-2 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 11382027 06/3142 5 4 3 2 1 0

Lecture Notes in Artificial Intelligence 3345
Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Lecture Notes in Artificial Intelligence (LNAI)

Vol. 3345: Y. Cai (Ed.), Ambient Intelligence for Scientific Discovery. XIV, 311 pages. 2005.

Vol. 3339: G.I. Webb, X. Yu (Eds.), AI 2004: Advances in Artificial Intelligence. XXII, 1272 pages. 2004.

Vol. 3336: D. Karagiannis, U. Reimer (Eds.), Practical Aspects of Knowledge Management. X, 523 pages. 2004.

Vol. 3327: Y. Shi, W. Xu, Z. Chen (Eds.), Data Mining and Knowledge Management. XIII, 263 pages. 2004.

Vol. 3315: C. Lemaître, C.A. Reyes, J.A. González (Eds.), Advances in Artificial Intelligence – IBERAMIA 2004. XX, 987 pages. 2004.

Vol. 3303: J.A. López, E. Benfenati, W. Dubitzky (Eds.), Knowledge Exploration in Life Science Informatics. X, 249 pages. 2004.

Vol. 3275: P. Perner (Ed.), Advances in Data Mining. VIII, 173 pages. 2004.

Vol. 3265: R.E. Frederking, K.B. Taylor (Eds.), Machine Translation: From Real Users to Research. XI, 392 pages. 2004.

Vol. 3264: G. Paliouras, Y. Sakakibara (Eds.), Grammatical Inference: Algorithms and Applications. XI, 291 pages. 2004.

Vol. 3259: J. Dix, J. Leite (Eds.), Computational Logic in Multi-Agent Systems. XII, 251 pages. 2004.

Vol. 3257: E. Motta, N.R. Shadbolt, A. Stutt, N. Gibbins (Eds.), Engineering Knowledge in the Age of the Semantic Web. XVII, 517 pages. 2004.

Vol. 3249: B. Buchberger, J.A. Campbell (Eds.), Artificial Intelligence and Symbolic Computation. X, 285 pages. 2004.

Vol. 3245: E. Suzuki, S. Arikawa (Eds.), Discovery Science. XIV, 430 pages. 2004.

Vol. 3244: S. Ben-David, J. Case, A. Maruoka (Eds.), Algorithmic Learning Theory. XIV, 505 pages. 2004.

Vol. 3238: S. Biundo, T. Frühwirth, G. Palm (Eds.), KI 2004: Advances in Artificial Intelligence. XI, 467 pages. 2004.

Vol. 3230: J.L. Vicedo, P. Martínez-Barco, R. Muñoz, M. Saiz Noeda (Eds.), Advances in Natural Language Processing. XII, 488 pages. 2004.

Vol. 3229: J.J. Alferes, J. Leite (Eds.), Logics in Artificial Intelligence. XIV, 744 pages. 2004.

Vol. 3228: M.G. Hinchey, J.L. Rash, W.F. Truszkowski, C.A. Rouff (Eds.), Formal Approaches to Agent-Based Systems. VIII, 290 pages. 2004.

Vol. 3215: M.G.. Negoita, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part III. LVII, 906 pages. 2004.

Vol. 3214: M.G.. Negoita, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part II. LVIII, 1302 pages. 2004.

Vol. 3213: M.G.. Negoita, R.J. Howlett, L.C. Jain (Eds.), Knowledge-Based Intelligent Information and Engineering Systems, Part I. LVIII, 1280 pages. 2004.

Vol. 3209: B. Berendt, A. Hotho, D. Mladenic, M. van Someren, M. Spiliopoulou, G. Stumme (Eds.), Web Mining: From Web to Semantic Web. IX, 201 pages. 2004.

Vol. 3206: P. Sojka, I. Kopecek, K. Pala (Eds.), Text, Speech and Dialogue. XIII, 667 pages. 2004.

Vol. 3202: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), Knowledge Discovery in Databases: PKDD 2004. XIX, 560 pages. 2004.

Vol. 3201: J.-F. Boulicaut, F. Esposito, F. Giannotti, D. Pedreschi (Eds.), Machine Learning: ECML 2004. XVIII, 580 pages. 2004.

Vol. 3194: R. Camacho, R. King, A. Srinivasan (Eds.), Inductive Logic Programming. XI, 361 pages. 2004.

Vol. 3192: C. Bussler, D. Fensel (Eds.), Artificial Intelligence: Methodology, Systems, and Applications. XIII, 522 pages. 2004.

Vol. 3191: M. Klusch, S. Ossowski, V. Kashyap, R. Unland (Eds.), Cooperative Information Agents VIII. XI, 303 pages. 2004.

Vol. 3187: G. Lindemann, J. Denzinger, I.J. Timm, R. Unland (Eds.), Multiagent System Technologies. XIII, 341 pages. 2004.

Vol. 3176: O. Bousquet, U. von Luxburg, G. Rätsch (Eds.), Advanced Lectures on Machine Learning. IX, 241 pages. 2004.

Vol. 3171: A.L.C. Bazzan, S. Labidi (Eds.), Advances in Artificial Intelligence – SBIA 2004. XVII, 548 pages. 2004.

Vol. 3159: U. Visser, Intelligent Information Integration for the Semantic Web. XIV, 150 pages. 2004.

Vol. 3157: C. Zhang, H. W. Guesgen, W.K. Yeap (Eds.), PRICAI 2004: Trends in Artificial Intelligence. XX, 1023 pages. 2004.

Vol. 3155: P. Funk, P.A. González Calero (Eds.), Advances in Case-Based Reasoning. XIII, 822 pages. 2004.

Vol. 3139: F. Iida, R. Pfeifer, L. Steels, Y. Kuniyoshi (Eds.), Embodied Artificial Intelligence. IX, 331 pages. 2004.

Vol. 3131: V. Torra, Y. Narukawa (Eds.), Modeling Decisions for Artificial Intelligence. XI, 327 pages. 2004.

Vol. 3127: K.E. Wolff, H.D. Pfeiffer, H.S. Delugach (Eds.), Conceptual Structures at Work. XI, 403 pages. 2004.

Vol. 3123: A. Belz, R. Evans, P. Piwek (Eds.), Natural Language Generation. X, 219 pages. 2004.

Vol. 3120: J. Shawe-Taylor, Y. Singer (Eds.), Learning Theory. X, 648 pages. 2004.

Vol. 3097: D. Basin, M. Rusinowitch (Eds.), Automated Reasoning. XII, 493 pages. 2004.

Vol. 3071: A. Omicini, P. Petta, J. Pitt (Eds.), Engineering Societies in the Agents World. XIII, 409 pages. 2004.

Vol. 3070: L. Rutkowski, J. Siekmann, R. Tadeusiewicz, L.A. Zadeh (Eds.), Artificial Intelligence and Soft Computing - ICAISC 2004. XXV, 1208 pages. 2004.

Vol. 3068: E. André, L. Dybkjær, W. Minker, P. Heisterkamp (Eds.), Affective Dialogue Systems. XII, 324 pages. 2004.

Vol. 3067: M. Dastani, J. Dix, A. El Fallah-Seghrouchni (Eds.), Programming Multi-Agent Systems. X, 221 pages. 2004.

Vol. 3066: S. Tsumoto, R. Słowiński, J. Komorowski, J.W. Grzymała-Busse (Eds.), Rough Sets and Current Trends in Computing. XX, 853 pages. 2004.

Vol. 3065: A. Lomuscio, D. Nute (Eds.), Deontic Logic in Computer Science. X, 275 pages. 2004.

Vol. 3060: A.Y. Tawfik, S.D. Goodwin (Eds.), Advances in Artificial Intelligence. XIII, 582 pages. 2004.

Vol. 3056: H. Dai, R. Srikant, C. Zhang (Eds.), Advances in Knowledge Discovery and Data Mining. XIX, 713 pages. 2004.

Vol. 3055: H. Christiansen, M.-S. Hacid, T. Andreasen, H.L. Larsen (Eds.), Flexible Query Answering Systems. X, 500 pages. 2004.

Vol. 3048: P. Faratin, D.C. Parkes, J.A. Rodríguez-Aguilar, W.E. Walsh (Eds.), Agent-Mediated Electronic Commerce V. XI, 155 pages. 2004.

Vol. 3040: R. Conejo, M. Urretavizcaya, J.-L. Pérez-de-la-Cruz (Eds.), Current Topics in Artificial Intelligence. XIV, 689 pages. 2004.

Vol. 3035: M.A. Wimmer (Ed.), Knowledge Management in Electronic Government. XII, 326 pages. 2004.

Vol. 3034: J. Favela, E. Menasalvas, E. Chávez (Eds.), Advances in Web Intelligence. XIII, 227 pages. 2004.

Vol. 3030: P. Giorgini, B. Henderson-Sellers, M. Winikoff (Eds.), Agent-Oriented Information Systems. XIV, 207 pages. 2004.

Vol. 3029: B. Orchard, C. Yang, M. Ali (Eds.), Innovations in Applied Artificial Intelligence. XXI, 1272 pages. 2004.

Vol. 3025: G.A. Vouros, T. Panayiotopoulos (Eds.), Methods and Applications of Artificial Intelligence. XV, 546 pages. 2004.

Vol. 3020: D. Polani, B. Browning, A. Bonarini, K. Yoshida (Eds.), RoboCup 2003: Robot Soccer World Cup VII. XVI, 767 pages. 2004.

Vol. 3012: K. Kurumatani, S.-H. Chen, A. Ohuchi (Eds.), Multi-Agents for Mass User Support. X, 217 pages. 2004.

Vol. 3010: K.R. Apt, F. Fages, F. Rossi, P. Szeredi, J. Váncza (Eds.), Recent Advances in Constraints. VIII, 285 pages. 2004.

Vol. 2990: J. Leite, A. Omicini, L. Sterling, P. Torroni (Eds.), Declarative Agent Languages and Technologies. XII, 281 pages. 2004.

Vol. 2980: A. Blackwell, K. Marriott, A. Shimojima (Eds.), Diagrammatic Representation and Inference. XV, 448 pages. 2004.

Vol. 2977: G. Di Marzo Serugendo, A. Karageorgos, O.F. Rana, F. Zambonelli (Eds.), Engineering Self-Organising Systems. X, 299 pages. 2004.

Vol. 2972: R. Monroy, G. Arroyo-Figueroa, L.E. Sucar, H. Sossa (Eds.), MICAI 2004: Advances in Artificial Intelligence. XVII, 923 pages. 2004.

Vol. 2969: M. Nickles, M. Rovatsos, G. Weiss (Eds.), Agents and Computational Autonomy. X, 275 pages. 2004.

Vol. 2961: P. Eklund (Ed.), Concept Lattices. IX, 411 pages. 2004.

Vol. 2953: K. Konrad, Model Generation for Natural Language Interpretation and Analysis. XIII, 166 pages. 2004.

Vol. 2934: G. Lindemann, D. Moldt, M. Paolucci (Eds.), Regulated Agent-Based Social Systems. X, 301 pages. 2004.

Vol. 2930: F. Winkler (Ed.), Automated Deduction in Geometry. VII, 231 pages. 2004.

Vol. 2926: L. van Elst, V. Dignum, A. Abecker (Eds.), Agent-Mediated Knowledge Management. XI, 428 pages. 2004.

Vol. 2923: V. Lifschitz, I. Niemelä (Eds.), Logic Programming and Nonmonotonic Reasoning. IX, 365 pages. 2003.

Vol. 2915: A. Camurri, G. Volpe (Eds.), Gesture-Based Communication in Human-Computer Interaction. XIII, 558 pages. 2004.

Vol. 2913: T.M. Pinkston, V.K. Prasanna (Eds.), High Performance Computing - HiPC 2003. XX, 512 pages. 2003.

Vol. 2903: T.D. Gedeon, L.C.C. Fung (Eds.), AI 2003: Advances in Artificial Intelligence. XVI, 1075 pages. 2003.

Vol. 2902: F.M. Pires, S.P. Abreu (Eds.), Progress in Artificial Intelligence. XV, 504 pages. 2003.

Vol. 2892: F. Dau, The Logic System of Concept Graphs with Negation. XI, 213 pages. 2003.

Vol. 2891: J. Lee, M. Barley (Eds.), Intelligent Agents and Multi-Agent Systems. X, 215 pages. 2003.

Vol. 2882: D. Veit, Matchmaking in Electronic Markets. XV, 180 pages. 2003.

Vol. 2872: G. Moro, C. Sartori, M.P. Singh (Eds.), Agents and Peer-to-Peer Computing. XII, 205 pages. 2004.

Vol. 2871: N. Zhong, Z.W. Raś, S. Tsumoto, E. Suzuki (Eds.), Foundations of Intelligent Systems. XV, 697 pages. 2003.

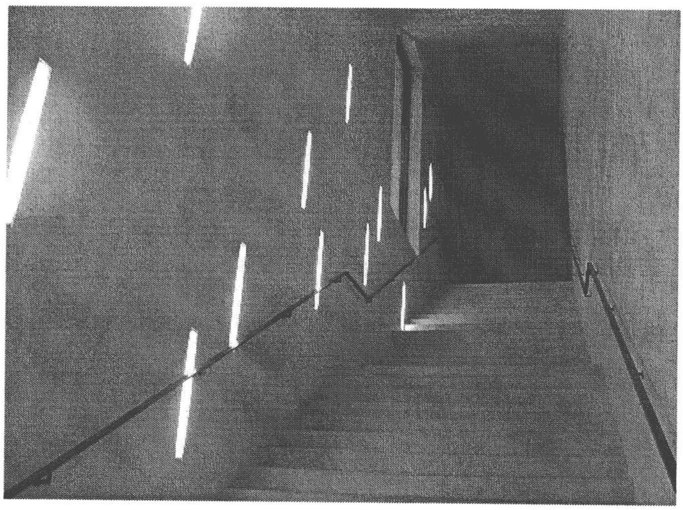
Vol. 2854: J. Hoffmann, Utilizing Problem Structure in Planing. XIII, 251 pages. 2003.

Vol. 2843: G. Grieser, Y. Tanaka, A. Yamamoto (Eds.), Discovery Science. XII, 504 pages. 2003.

Vol. 2842: R. Gavaldá, K.P. Jantke, E. Takimoto (Eds.), Algorithmic Learning Theory. XI, 313 pages. 2003.

Vol. 2838: N. Lavrač, D. Gamberger, L. Todorovski, H. Blockeel (Eds.), Knowledge Discovery in Databases: PKDD 2003. XVI, 508 pages. 2003.

Vol. 2837: N. Lavrač, D. Gamberger, L. Todorovski, H. Blockeel (Eds.), Machine Learning: ECML 2003. XVI, 504 pages. 2003.



The Museum of Modern Art is perhaps one of the most interesting landmarks in Vienna, where SIGCHI 2004 Workshop of "Ambient Intelligence for Scientific Discovery" was held. The building is covered by walls and ceiling composed of thick slabs of dark basalt stone cut at intervals by narrow vertical slits, accentuating the monolithic effect of the entire volume.

Preface

For half a century, computer scientists have been working on systems for discovering lawful patterns in letters, numbers, words and images. The research has expanded into the computational study of the process of scientific discovery, producing such well-known AI programs as BACON and DENDRAL. However, *autonomous* discovery systems have been rarely used in the real world. While many factors have contributed to this, the most chronic difficulties seem always to fall into two categories: (1) the representation of the prior knowledge that people bring to their tasks, and (2) the awareness of new context.

Many difficult scientific discovery tasks can only be solved in *interactive* ways, by combining intelligent computing techniques with intuitive and adaptive user interfaces. It is inevitable that human intelligence is used in scientific discovery systems. For example, the human eyes can capture complex patterns and relationships, along with detecting the exceptional cases in a data set. The human brain can easily manipulate perceptions (shape, color, balance, time, distance, direction, speed, force, similarity, likelihood, intent and well-being) to make decisions. This process consists of *perception* and *communication* and it is often ubiquitous and autonomous. We refer to this kind of intelligence as ambient intelligence (AmI).

Ambient intelligence is about human interaction with information in a way that permits humans to spot interesting signs in massive data sources – building tools that capitalize on human strengths and compensate for human weaknesses to enhance and extend discovery capabilities. For example, people are much better than machines at detecting patterns in a visual scene, while machines are better at manipulating streams of numbers.

Scientific discovery is a process of creative perception and communication. With growing data streams and the complexity of discovery tasks, we see a demand for integrating novel digital media and communications (e.g., body media, capsule cameras, WiFi, etc.) and opportunities for ambient intelligence to use interaction methods that are usually taken for granted, such as perception, insight and analogy. We want to search for solutions to interesting questions such as: How do we significantly reduce information while maintaining meaning? How do we extract patterns from massive and growing data resources?

This volume represents the outcome of the SIGCHI Workshop on "Ambient Intelligence for Scientific Discovery," held in Vienna, on April 25, 2004. The chapters in this volume were selected from the revised papers submitted to the workshop and contributions from leading researchers in this area. The objective of this volume is to present a state-of-the-art survey of studies in ambient intelligence for scientific discovery, including novel ideas, insightful findings and ambient intelligence systems across multiple disciplines and applications. The volume is published for graduate students, senior undergraduate students, researchers and professionals. Therefore, extended references are provided in each chapter.

The contents in this volume are organized into three tracks: Part I, New Paradigms in Scientific Discovery; Part II, Ambient Cognition; and Part III, Ambient Intelligence Systems. Many chapters share common features such as interaction, vision, language, and biomedicine, which reflect the interdisciplinary nature of this volume.

I. New Paradigms in Scientific Discovery. Processing massive data has been a bottleneck to modern sciences. In Chap. 1, "Science at the Speed of Thought," Devaney et al. describe a virtual laboratory that is designed to accelerate scientific exploration and discovery by minimizing the time between the generation of a scientific hypothesis and the test of that idea, and thereby enabling science at the speed of thought. In Chap. 2, "Computational Biology and Language," Ganapathiraju et al. present a breakthrough approach that enables exploitation of an analogy between natural language and speech processing techniques in computational biology. In Chap. 3, "Interactive Comprehensible Data Mining," Pryke and Beale present their interactive data mining system that helps users gain insight from the dynamically created virtual data space. In Chap. 4, "Scientific Discovery Within Data Streams," Cowell et al. present the architecture of a next-generation analytical environment for scientific discovery within continuous, time-varying data streams.

II. Ambient Cognition. Understanding how people sense, understand and use images and words in everyday work and life can eventually help us design more effective discovery systems. In Chap. 5, Leyton reviews his theory of "Shape as Memory Storage", addressing shape description over time. Leyton's theory has been used in more than 40 fields, such as radiology, metrology, computer vision, chemical engineering, geology, computer-aided design, anatomy, botany, forensic science, software engineering, architecture, linguistics, mechanical engineering, computer graphics, art, semiotics, archaeology, and anthropology, etc. In Chap. 6, Hubona and Shirah investigate how various spatial depth cues, such as motion, stereoscopic vision, shadows and scene background, are utilized to promote the perceptual discovery and interpretation of the presented imagery in 3D scientific visualization. In Chap. 7, "Textual Genre Analysis and Identification," Kaufer et al. present a knowledge-based approach for encoding a large library of English strings used to capture textual impressions and report on a study of a popular textual genre – the technology review. The expert system incorporates contextual information, e.g., culture, emotion, context, and purpose, etc., which is different from many prevailing methods such as machine learning or statistics. In Chap. 8, "Cognitive Artifacts in Complex Work," Jones and Nemeth use acute care and scientific ethnographic field studies to show how cognitive artifacts can be used to grasp the nature of cognitive work in uncertain, complex, technical work settings. This front-end research is aimed at optimizing the distributed cognitive work.

III. Ambient Intelligence Systems. Ubiquitous sensors and communication technologies not only can assist scientific discovery, but can also catalyze new sciences. In Chap. 9, "Multi-modal Interaction in Biomedicine," Zudilova and Sloot investigate the practical deployment of virtual reality systems in the medical environment. They explore the multi-modal interaction of virtual reality and desktop computers in Virtual Radiology Explorer. In Chap. 10, "Continuous Body Monitoring," Farringdon and Nashold describe a personal and continuous body monitor that is one of the few examples of ambient intelligence devices commercially available today. This also brings challenges to sciences: for example, how do we extract the interesting patterns from a continuous body monitor? From this example we can see how the research scope has been extended from laboratories to homes and in vivo. In Chap. 11, "Ambient Diagnosis," Cai et al. explore *Ambient Diagnosis* that is based on traditional Chinese medicine (TCM). The case study shows how to map the visual features on the tongue into a vector of numbers. In Chap. 12, Tanz et al. describe methods for location mapping in a wireless local area network (WLAN) and applications in social sciences. The system cmuSKY developed by the authors has become a public online resource for scientific discovery. In Chap. 13, "Behavior-Based Indoor Navigation," Abascal et al. present a method for motor fusion using ambient information from the environment. Indoor robotic navigation has been an active subject because of applications in assisted-living, such as smart wheelchair control, guidance for the visually impaired, or indoor assistance of the elderly. Finally, in Chap. 14, "Ambient Intelligence Through Agile Agents," O'Hare et al. explore agile agents as a key enabler for the realization of the ambient intelligence vision.

We are deeply indebted to all the authors who contributed papers to this volume; without this depth of support and commitment there would have been no meaningful product at all. We acknowledge the members of the program committee, all those involved in the refereeing process, the workshop organizers, and all those in the community who helped to convene a successful workshop. Special thanks go to Judith Klein-Seetharaman, Peter Jones, William Eddy, David Kaufer, Yongxiang Hu, Bin Lin, and Vijayalaxmi Manoharan for their contributions to the workshop and this volume. Thanks to the external reviewers Howard Choset, Lori Levin, Susan Fussell, and Tony Adriaansen for their comments on the manuscripts. Thanks to Teri Mick for assisting the volume editing and Sarah Nashold for assisting the book design. This project is in part sponsored by NASA grant NAG-1-03024 and National Academy of Sciences (NAS) grant T-37.

Pittsburgh, October 2004

Yang Cai

Organization Committee and Reviewers

Yang Cai (Co-chair), Carnegie Mellon University, USA
Judith Klein-Seetharaman (Co-chair), Carnegie Mellon University, University of Pittsburgh, USA, and Forschungszentrum Juelich, Germany
Peter Jones, Redesign Research, USA
Elena Zudilova, University of Amsterdam, Netherlands
Yongxiang Hu, NASA Langley Research Center, USA
Bin Lin, NASA Langley Research Center, USA
Lori Levin, Carnegie Mellon University, USA
Gregory O'Hare, University College Dublin, Ireland
Howard Choset, Carnegie Mellon University, USA
Judith Devaney, National Institute of Standards and Technology, USA
Tony Adriaansen, Telecommunications & Industrial Physics, CSIRO, Australia
Susan Fussell, Carnegie Mellon University, USA

Sponsoring Institutions

National Aeronautics and Space Administration (NASA), USA
National Academy of Sciences (NAS), USA

Table of Contents

Part I: New Paradigms in Scientific Discovery

Part II: Ambient Cognition

Part III: Ambient Intelligence Systems

Science at the Speed of Thought

Judith E. Devaney[1], S.G. Satterfield[1], J.G. Hagedorn[1], J.T. Kelso[1], A.P. Peskin[1], W.L. George[1], T.J. Griffin[1], H.K. Hung[1], and R.D. Kriz[2]

[1] National Institute of Standards and Technology, USA
judith.devaney@nist.gov
[2] Virginia Tech, USA
rkriz@vt.edu

1 Introduction

Scientific discoveries occur with iterations of theory, experiment, and analysis. But the methods that scientists use to go about their work are changing [1].

Experiment types are changing. Increasingly, experiment means computational experiment [2], as computers increase in speed, memory, and parallel processing capability. Laboratory experiments are becoming parallel as combinatorial experiments become more common.

Acquired datasets are changing. Both computer and laboratory experiments can produce large quantities of data where the time to analyze data can exceed the time to generate it. Data from experiments can come in surges where the analysis of each set determines the direction of the next experiments. The data generated by experiments may also be non-intuitive. For example, nanoscience is the study of materials whose properties may change greatly as their size is reduced [3]. Thus analyses may benefit from new ways to examine and interact with data.

Two factors will accelerate these trends and result in increasing volumes of data:

- CPU speedup: as companies strive to keep Moore's law [4,5] in effect
- Computer architecture speedup: as all computers benefit from architecture advances in high end computers.

Figure 1 gives an overview of how these impact problems [6]. These factors make computers ever more capable and increase the move to computational experiments and automation.

But a third factor offers a partial solution: graphics speedup. Computer game enthusiasts are funding a fast pace of development of graphics processing units (GPUs) [7,8]. The use of these GPUs in the support of science makes the future world increasingly computational, visual, and interactive.

We believe that representation and interaction drive discovery, and that bringing the experiments (computer and laboratory) of science into an interactive, immersive, and collaborative environment provides opportunities for speed and synergy. Adding traditional data analysis, machine learning, and data mining tools, with multiple representations and interactions can speed up the rate

Y. Cai (Ed.): Ambient Intelligence for Scientific Discovery, LNAI 3345, pp. 1–24, 2005.

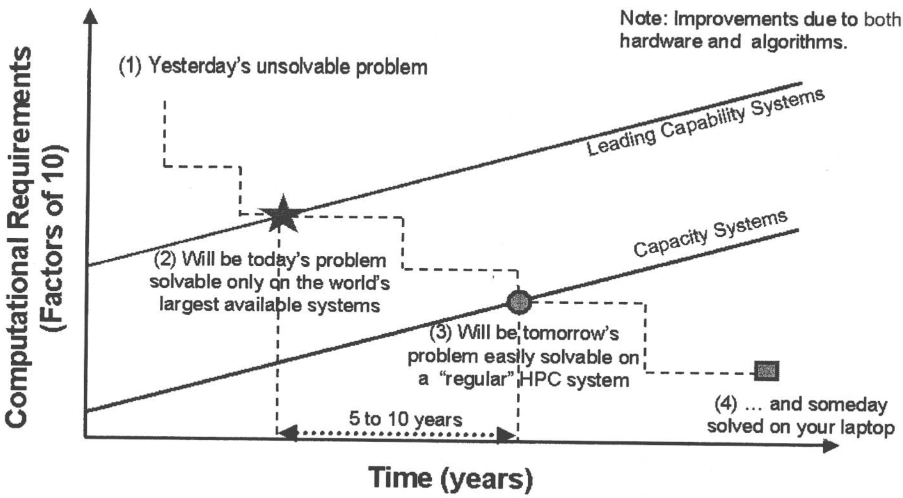


Fig. 1. Transition of solving important problems from unsolvable to solvable on your laptop.

of exploration and lead to new insights and discovery. Creating an environment that is efficient, general and flexible enough to work well across a wide variety of scientific applications is at the heart of our Virtual Laboratory (VL) design.

In section 2, we describe the VL we are building at NIST to address these issues. In section 3, we describe some applications. We present conclusions and future work in section 4.

2 The Virtual Laboratory

The VL needs to be efficient, general, and flexible, but it also needs to be able to get applications into it quickly in order to speed up the process of science and not burden it. Representations and interactions of many types need to be available and easily accessed. To accomplish this, our design consists of the following components:

- A distributed computing environment that provides the communication fabric of the VL,
- An immersive visualization environment that provides representation, interaction, and collaboration capability in the VL,
- A suite of tools for machine learning and analysis.

We will discuss each of these in turn.

2.1 Distributed Computing Environment

An important part of the VL is the capability of users to interact with their data sources, analysis programs, and their experiments, either computer experiments or laboratory experiments, from any of the supported VL access points (see Fig. 2

for a schematic). Examples of access points are the immersive visualization system or a remotely connected workstation. For the purposes of this discussion, data sources and analysis programs will be subsumed under computer experiment. The interactions range from viewing the status of their currently running experiment, or the results of a prior experiment, to providing feedback to the experiment in order to alter or restart the experiment. To provide this range of access requires a framework that enables communication between the user, the experiment, the visualization, and possibly with other collaborators actively interacting with the experiment.

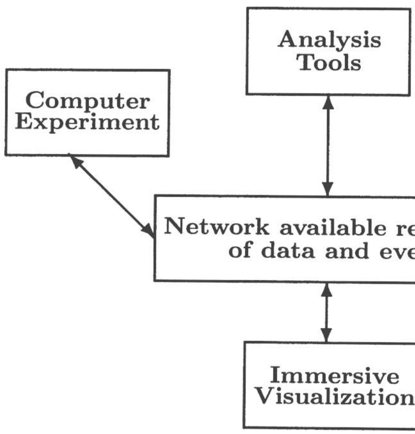


Fig. 2. Schematic of Distributed Computing Environment. Each component may also access other data sources.

The VL uses what is generically referred to as a coordination framework to provide a loose coupling between all participating entities of an experiment. The coupling is loose in that participants, that is, laboratory experiment equipment, computer simulations, visualization systems, user interfaces, and any other connection to the VL, can choose whether or not to utilize this coupling and can connect and disconnect from it at any time without disrupting the system. The VL distributed computing framework is implemented using the Java technology of Jini and JavaSpaces [9] [10]. Both of these packages, Jini and JavaSpaces, are available as pure Java implementations and so are portable to any system that supports a Java virtual machine, which includes systems running Linux, Microsoft Windows, and most Unix based operating systems.

Jini provides support for a form of distributed computing that explicitly handles common issues that arise in a distributed (networked) system, such as intermittent network outages and server crashes. Support includes automatic discovery of services available on the network, active leasing of services to help maintain current service information and to purge services that no longer exist, and distribution of events to remote applications to allow applications to communicate asynchronously and to react to expected or unexpected developments.

JavaSpaces is a specification for a Jini service that provides a coordination framework in the form of a tuple space. The concept of a tuple space was first introduced in the early 1980s by computer scientists David Gelernter and Nick Carriero, within the context of the programming language Linda [11] [12]. This is referred to as a coordination framework since it allows a loose collection of applications, linked over a common network, to communicate asynchronously. This communication is so loosely coupled that the applications do not need to be running, or even exist, at the same time. To take the possibilities to an extreme, application A can send a message that is ultimately to be read by application B (and possibly others) which has not yet been written and will run on a machine that has not yet been built. When application B receives the message, application A, and the machine it ran on, may no longer exist.

There are many ways to describe the purpose and use of a JavaSpace. The concept of a coordination framework is a good high-level description. At a lower level, a JavaSpace can be thought of simply as a shared memory space, accessible from any machine on the network and addressed using an associative lookup rather than by memory address. Objects stored in a JavaSpace are instances of classes in Java that have a few special characteristics needed to support storage and retrieval from the JavaSpace. So the associative lookup uses the Java class type system to identify objects to be written or read from the JavaSpace. This is a very robust addressing scheme, compared to using raw memory addressing or simple string matching, since it ensures that you receive an object of the correct type when you read from the space. These objects can store any type of information that might be needed by the applications. It is also possible to maintain structures of objects in a JavaSpace, such as a linked list of objects, a tree of objects, or an array of objects. Objects can also simply be markers, holding absolutely no data, but giving information simply by their existence or non-existence in the JavaSpace.

In the VL, a JavaSpace is maintained to allow for the coordination of experiments, visualizations, and interacting collaborators accessing the VL through their workstation or other supported interface device. Typical information stored in the VL JavaSpace includes experiment parameters, current status of an experiment such at the latest time-step of a computer simulation, or latest result from a laboratory experiment.

Of course, applications need not be written in Java to participate in the VL. In fact, it is expected that most computer simulations will likely be written in Fortran or C/C++, and most participating laboratory devices will likely not have direct network access. In Jini, applications and devices that are not capable of participating directly, either because of insufficient resources to run a Java virtual machine, or because they are closed systems, can still participate through the use of a surrogate [13]. These surrogates allow the application or device to participate by performing as a communications gateway to the Jini/JavaSpace network and also performing any computation needed in the process. The communication between the surrogate and the Jini/JavaSpace network uses the standard Jini/JavaSpaces protocols. The communication between the surrogate and