Ken Kahn

42 Medcroft Road, Tackley OX5 3AH, UK
Voice: +44 (0)1869 331 621; Fax: +1 (603) 962-8566
Email: KenKahn@ToonTalk.com

Educational Background:

Massachusetts Institute of Technology, Sept. 1973 to Feb. 1979
Ph.D. in Electrical Engineering and Computer Science, Jan. 1979
Thesis: *Creation of Computer Animation from Story Descriptions*
M.S. in Electrical Engineering, Aug. 1975
Thesis: *Mechanization of Temporal Knowledge*

University of Pennsylvania, Sept. 1969 to June 1973; University of Stockholm, Junior Year Abroad 1971
B.A. in Economics *magna cum laude* with distinction, June 1973
Thesis: *Bankruptcy, Information Costs, and Equilibrium*

Professional Experience:

Senior Researcher, Learning Technologies Group, Oxford University Computing Services. May 2006 to present

Visiting Fellow, consultant, and developer at the London Knowledge Lab, Institute of Education. June 2003 to present

Founder and President, Animated Programs, September 1992 to present

Visiting Research Scientist, Royal Institute of Technology, Sweden. Sept. 2002 to Sept. 2003; July 2004 to September 2005

Leverhulme Trust Fellowship at the Institute of Education, London, Sept. 2003 through June 2004

Consultant to the University of Stockholm, Sweden, November 1999 to August 2001

Consultant to the Institute of Education, University of London, November 1998 to August 2001

Consultant to Uppsala University, Sweden, November 1998 to November 1999

Visiting Scholar, CSLI, Stanford University, September 1992 to September 1996

Independent consultant to IBM and Swedish Institute of Computer Science, 1995

Member of Research Staff, Xerox Palo Alto Research Center (PARC), August 1984 to August 1992

Guest Researcher, ICOT (Institute for New Generation Computer Technology), Tokyo, November 1983 and June 1992

Consultant for Rand Corporation, Atari Research, Syracuse University, and CitiCorp, 1982 to 1983

Associate Professor, Computing Science Department, Uppsala University, July 1981 to August 1984

Research Associate, UPMAIL, Computing Science Dept. Uppsala University, July 1980 to August 1984

Visiting Professor and Researcher, University of Stockholm, January 1980 to June 1980

MIT Research Scientist and Lecturer, January 1979 to December 1979

Research Assistant, Research Staff, and IBM Graduate Student Fellowship Recipient, Artificial Intelligence Laboratory, Project MAC (Laboratory for Computer Science), MIT, September 1973 to January 1979

Programmer of statistical and graphical study of census data, Philadelphia Social History Project, Sept. 1972 to July 1973

Applications programmer, consultant and liaison to the humanities departments, University of Pennsylvania Computer Center, Jan. 1970 to July 1971

Research Experience:

As part of my master's thesis at MIT, I implemented two versions of a time specialist, a computer program capable of accepting a wide range of temporal statements, checking their consistency, and making inferences to answer questions [52, 2]. Following that I joined the LOGO group and taught elementary school children to do natural language programming [51] and computer animation [67] using several small systems I built for this purpose [1]. In 1983, while consulting for Atari, I resumed research on natural language tools for children, this time based upon Prolog [5].

From 1976 to 1980 I was the designer and implementer of the object-oriented computer language "Director" [23, 56, 60, 54]. Director was used for programming computer animation, knowledge-based systems, and experiments in programming by children. Simultaneously, I was working on my doctoral thesis, building a system capable of automatically creating simple computer-animated films on the basis of vague, incomplete story descriptions [53, 24, 26, 27, 55]. During this time, I took courses in animation and filmmaking (two at Harvard University) and made several films which were shown at film festivals, local theaters, and cable TV [67, 68, 70, 69]. In 1988 and 1989, as a refresher, I took three animation courses at De Anza College and made several short animated films.

After a year as a Lecturer and Research Scientist at MIT in 1979, I went to Sweden, initially to the University of Stockholm, and then to Uppsala University, where I began exploring multi-paradigm programming. After learning about Prolog, I used it to implement an actor language, "Intermission" [4]. Next came "Uniform", a language based on extended unification [8, 3]. The language was an attempt to combine the important features of Lisp, actor languages, and Prolog into a simple coherent framework. My research on combining the best of Lisp and Prolog led to my design and, with a colleague, implementation of LM-Prolog [57, 7, 58, 30, 31], an extended Prolog system on Lisp Machines that was sold by Lisp Machines, Incorporated.

While in Sweden I become excited about the potential for partial evaluation to win back performance sacrificed in the quest for simple generic programs. As a realistic example, I worked on automatically generating a compiler from LM-Prolog to Lisp by doing partial evaluation of the LM-Prolog interpreter written in Lisp [28, 29, 6]. The partial evaluator was written in LM-Prolog and generated efficient specializations of Lisp programs. It was during this period that my close interactions with the Japanese Fifth Generation Project began [14].

From Sweden I went to Xerox Palo Alto Research Center (PARC) to continue my research on multi-paradigm systems with the "LOOPS" group. During 1985, I was one of the two designers of CommonLoops [9], the basis of the Common Lisp Object System (CLOS) standard. At this time I was also chair of the Common Lisp object-oriented programming subcommittee. (Despite having been instrumental in bringing CLOS to the world, I am not particularly proud of it.) During this period, I also collaborated on other projects with several researchers at PARC. These ranged from CoLab (pioneering software support for collaboration) [12], virtual copies (a new approach to creating computational objects) [33], and integrating access-oriented programming with other programming paradigms [10].

In 1986 I started and led the Vulcan project, whose purpose was building high level programming abstractions within a concurrent logic programming framework [32, 59, 11, 34, 60, 62, 61]. For nearly two years I managed a project consisting of six researchers. Much of the research focused upon programming language support for building distributed systems [13]. When funding was cut, similar efforts were continued, though on a smaller scale [40, 64, 37, 35]. I focused upon connections between concurrent logic (and more generally constraint) programming and concurrent object-oriented frameworks or actors [63, 38, 36]. The theme underlying Vulcan and subsequent research was "clean and small but real".

From 1989 to 1992, I combined my interest in animation with my interests in language design and concurrency by leading a project to visualize concurrent programs and animate their executions in a coherent and general manner [39, 71]. My efforts resulted in a picture parser and an animator capable of interpreting PostScript drawings of concurrent constraint programs and automatically producing animations

of program executions [41]. I explored applications of this system to visualizing object-oriented designs [66], real-time control, and non-programming uses (e.g. workflow and animated proofs [65]).

In September 1992 I left Xerox PARC to found Animated Programs, a company whose mission is to bring the power and fun of programming to children and adults. I am the sole designer and developer of ToonTalk, a concurrent animated programming language [42, 43, 15, 16, 44, 17, 45, 46, 18, 47, 19, 21, 50]. I have presented ToonTalk at a dozen conferences on game development and educational software and have advised doctoral students [20, 49, 72] whose research builds upon ToonTalk.

I have participated in two EU-funded educational research projects building upon ToonTalk [48, 22, 73]: Playground, which supported very young children in making their own computer games; and WebLabs, where children build scientific models, explore mathematics computationally, and publish reports with runnable models on the Web. In both projects my role included supporting and enhancing the technologies underlying the research, contributing to project design and planning, building tools and components for use by both children and researchers, helping children use these tools to build games or to explore science and mathematics, and testing the tools and learning material with children.

I taught elementary school children to program graphics, animation, and natural language while a member of the Logo Group of the MIT AI Laboratory in 1975. Since 1995 I have been teaching ToonTalk to children. I taught fourth grade students at the Encinal School in Menlo Park, California for several hours a week for three years. I taught a diverse group of children at Plugged In Greenhouse (a creative arts and technology studio for low income children between 6 to 12 years old) in East Palo Alto, California. I taught ToonTalk workshops in museums in San Jose, California and Tokyo, Japan. Most recently, I have worked with children at the Theydon Bois Primary School in Essex and the Cherwell School in Oxford on exploring infinite sets in ToonTalk as part of the WebLabs Project.

From September 2005 to March 2006 I was the primary software designer and developer in the London Knowledge Lab/MIT Media Lab project funded by the BBC Digital Curriculum Project (renamed to Jam). Within this project I designed and built two game construction kits for learning mechanics and an associated meta game for assisting in the learning and game construction [77]. Since May 2006 I have been leading The Constructing2Learn Project at Oxford University [78,79] where we are developing tools and learning designs to enable non-programmers to build serious computer models. I will be leading the 2-year Modelling4All Project due to start August 2007. Since May 2006 I have also been a member of the London Knowledge Lab's participation in the European ReMath Project where we have developed a construction kit called MoPiX based upon algebra. Beginning October 2007, I'll join the London Knowledge Lab MiGen Project part-time.

University Teaching Experience:

Visiting Lecturer for a graduate course on Visual Programming at the Royal Institute of Technology, Sweden. Spring 2000.

Lecturer for undergraduate and graduate courses in Artificial Intelligence, Description Languages, Lisp Machines, and Logic Programming, Uppsala University, Sweden, Sept. 1980 to August 1984.

Lecturer for two courses in Artificial Intelligence, University of Stockholm, Sweden, Jan. to June 1980.

Lecturer for "Structure and Interpretation of Computer Languages", Department of Electrical Engineering and Computer Science, MIT, Feb. 1979 to June 1979.

Teaching Assistant for "Problem-Solving Paradigms", Department of Electrical Engineering and Computer Science, MIT, January 1976 to May 1976.

Graduate Student Thesis Advising:

I advised five students who completed their Ph.D. in computer science (one at Stanford University, two at Linköping University, one at Stockholm University, and one at Universidade de Trás-os-Montes e Alto Douro, Portugal). I have advised eight masters thesis students (MIT, University of Washington, and Stockholm University).

Editorial and Academic Activities:

Member of program committee: Second International Logic Programming Conference, Uppsala, Sweden, 1984; Third International Conference on Logic Programming, London, 1986; North American Conference on Logic Programming, Cleveland, Ohio, 1989; Joint Conference on Object-Oriented Programming and European Conference on Object-Oriented Programming, Ottawa, Canada, 1990; North American Conference on Logic Programming, Austin, Texas, 1990; Symposium on Partial Evaluation and Semantics-Based Program Manipulation, New Haven, Connecticut, 1991; International Logic Programming Symposium, San Diego, California, 1991. IEEE Conference on Visual Programming. Symposium on End User and Domain Specific Programming, 2002 and 2003.

General chair: International Logic Programming Symposium, San Diego, California, 1991.

Area editor: Knowledge Representation, Reasoning, and Expert Systems, *The Journal of Logic Programming*, 1986-1991.

Editorial board: *The Journal of Logic Programming, Higher-Order and Symbolic Computing* (current)*, New Generation Computing, IEEE Parallel and Distributed Technology: Systems and Applications*.

I have been on six doctoral thesis committees, and I have been the host/advisor for five undergraduate summer students, two graduate students, and two postdoctoral fellows while at Xerox PARC.

Patents:

United States Patent Number 5,517,663. Animated User Interface for Computer Program Creation, Control, and Execution. May 14, 1996.

Research Projects:

AI and medicine, MIT, 1973 to 1975.
LOGO Project, MIT, 1975 to 1977, 1979.
Actors and Open Systems Project, MIT, 1975 to 1979.
Programming Methodology and Artificial Intelligence, Uppsala University, 1980 to 1984.
LOOPS and CommonLoops Multi-paradigm Programming Project, Xerox PARC, 1984 to 1987.
Vulcan Distributed Programming Project, Xerox PARC, 1986 to 1990.
Janus Distributed Constraint Programming Project, Xerox PARC, 1989 to 1992.
Visual Janus Project, Xerox PARC 1990 to 1992.
Playground Project, EU, led by the Institute of Education, 1998 to 2001.
WebLabs Project, EU, led by the Institute of Education, 2002 to 2005.
BBC Digital Curriculum Project, London Knowledge Lab and MIT Media Lab, 2005 to 2006.
ReMath Project, EU, London Knowledge Lab, May 2006 to present.
Constructing2Learn Project, JISC, Oxford University Computing Services, May 2006 to November 2007.

Recent invited lectures and seminars:

May 2007, Swedish Institute of Computer Science, Kista, Sweden.
February 2007, Centre for Applied Research in Educational Technologies at Cambridge University
January 2007, University of Surrey, Sociology Department, UK.
November 2006, Oxford University Complex Systems Seminar, UK.
November 2006, Plenary, ISSEP 2006, Vilnius, Lithuania.
October 2006, Oxford University Zoology Department, UK.
April 2006, London Knowledge Lab, University of London, UK.
March 2006, Oxford Brookes University, Department of Computing, Oxford, UK.
Jan. 2006, Oxford University E-Learning seminar, Department of Educational Studies, UK.
Jan. 2006, University of Edinburgh, UK.
Nov. 2005, University of Stockholm Department of Education, Sweden.
Oct. 2005, University of Bratislava, Slovakia.
Oct. 2005, Knowledge Media Institute, Open University.
Sept 2005, University of Bristol.
June 2005, Psychology of Programming Interest Group 17th Workshop, invited keynote, Brighton, UK.
June 2005, Northwestern University, Evanston, USA.

June 2005, University of Warwick, UK.
May 2005, Microsoft Research Cambridge, UK.
April 2005, London Knowledge Lab, University of London, UK.
March 2005, University of Pennsylvania, Philadelphia, USA.
March 2005, SRI International, Menlo Park, CA, USA.
Feb. 2005, Oxford University E-Learning seminar, Department of Educational Studies, Oxford, UK.
Nov. 2004, Swedish Institute of Computer Science, Stockholm, Sweden.
Oct. 2004, Oxford University Computing Laboratory, Oxford, UK.
Sept. 2004, MIT Media Lab Europe, Dublin, Ireland.
July 2004, University of Canterbury Computer Science Department, Christchurch, New Zealand.
July 2004, Human Interface Technology Laboratory, Christchurch, New Zealand.
July 2004, Te Manawa Museum, Palmerston North, New Zealand.
June 2004, New Zealand Game Developers Conference, Invited speaker, Dunedin, New Zealand.
April 2004, Cambridge University Computing Laboratory, Cambridge, UK.
April 2004, IBM Research Center, Yorktown Heights, NY, USA.
June 2003, Linköping University Computer Science Department, Linköping, Sweden.
May 2003, Swedish HMI workshop, Söderköpings Brunn, Sweden.
May 2003, MIT Media Lab, Cambridge, MA, USA.
May 2003, MIT Laboratory for Computer Science, Cambridge, MA, USA.
July 2002, Waseda University Department of Computer Science, Tokyo, Japan.
June 2002, Keio University Department of Computer Science, Tokyo, Japan.
June 2002, Visual Programming Environment Symposium, Invited speaker, Tokyo, Japan.
March 2002, University of California at Berkeley Computer Science Department, Berkeley, CA, USA.
Aug. 2001, EuroLogo Conference, Keynote speaker, Linz, Austria.

Journal articles and book chapters:

[77] Ken Kahn, Richard Noss, Celia Hoyles, and Duncan Jones, Designing digital technologies for layered learning, *Informatics Education – The Bridge between Using and Understanding Computers*, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2006.

[74] Y. Mor, R. Noss, C. Hoyles, K. Kahn, G. Simpson, Designing to See and Share Structure in Number Sequences, *The International Journal for Technology in Mathematics Education,* Vol. 13, No. 2, 2006.

[22] Y. Mor, C. Hoyles, K. Kahn, R. Noss, and G. Simpson. Thinking in Process. *Micromath* 20/2, Association of Teachers of Mathematics, Summer 2004.

[21] Ken Kahn. ToonTalk – Steps Towards Ideal Computer-Based Learning Environments, in Mario Tokoro and Luc Steels, editors, *A Learning Zone of One's Own: Sharing Representations and Flow in Collaborative Learning Environments,* IOS Press Inc, June 2004.

[20] Leonel Morgado, Maria Gabriel Cruz, and Ken Kahn. ToonTalk in Kindergartens: Field Notes, in Mendez-Vilas, Antonio; Mesa González, José António; Solo de Zaldívar Maldonado, Inés (Eds.), "Information Society and Education - Proceedings of the International Conference on Information and Communication Technologies in Education (ICTE2002)", *Journal of Digital Contents*, Vol. 1, Issue 1: Formatex, Badajoz, Spain, 2003, ISSN: 1696-313X, ISBN: 84-607-8369-3

[19] Chris DiGiano, Ken Kahn, Allen Cypher, and David Canfield Smith. Integrating Learning Supports into the Design of Visual Programming Systems. *Journal of Visual Languages and Computing*. 12, 501-524. 2001.

[18] Ken Kahn. Generalizing by Removing Detail, *Communications of the ACM*, 43(3), March 2000. Extended version is in Henry Lieberman, editor, *Your Wish Is My Command: Programming By Example,* Morgan Kaufmann, 2001.

[17] Ken Kahn. Helping Children to Learn Hard Things: Computer Programming with Familiar Objects and Actions, in A. Druin, editor, *The Design of Children's Technology*, Morgan Kaufmann, 1998.

[16] Ken Kahn. Drawings on Napkins, Video Game Animation, and Other Ways of Programming Computers, *Communications of the ACM*, August 1996.

[15] Ken Kahn. ToonTalk™ -- An Animated Programming Environment for Children, *Proceedings of the National Educational Computing Conference*, Baltimore, Maryland, June 1995. Extended version in the *Journal of Visual Languages and Computing*, June 1996.

[14] Ken Kahn. A decade of progress in concurrent logic programming: A braid of research threads from ICOT, Xerox PARC, and Weizmann Institute. *Communications of the ACM*, 36(3), March 1993.

[13] Ken Kahn and Mark S. Miller. Language design and open systems. In B.A. Huberman, editor, *The Ecology of Computation*. North Holland, 1988.

[12] Mark Stefik, Gregg Foster, Daniel Bobrow, Ken Kahn, Stan Lanning, and Lucy Suchman. Beyond the chalkboard: Using computers to support collaboration and problem-solving meetings. *Communications of the ACM,* January 1987.

[11] K. Kahn, E. Tribble, M. Miller, and D. Bobrow. Vulcan: Logical concurrent objects. In B. Shriver and P. Wegner, editors, *Research Directions in Object-Oriented Programming*, pages 75--112. The MIT Press, 1987. Also in *Concurrent Prolog,* MIT Press, ed. Ehud Shapiro.

[10] Mark Stefik, Daniel Bobrow, and Ken Kahn. Integrating access-oriented programming into a multi-paradigm environment. *IEEE Software*, 3, 1986. Also in The Proceedings of the International Conference on System Science, Hawaii, 1986 and *AI Tools and Techniques*, ed. M. Richer, Ablex Publishing, 1989 and *Tutorial: Object-Oriented Computing*, Vol. 2, ed. G. Peterson, IEEE Press, 1987.

[9] Daniel Bobrow, Ken Kahn, Gregor Kiczales, Larry Masinter, Mark Stefik, and Frank Zdybel. CommonLoops: Merging Common Lisp and object-oriented programming. In *Proceedings of the ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications*, September 1986. Also In *Research Foundations in Object-Oriented and Semantic Database Systems*, eds. A. Cardenas and Dennis McLeod, Prentice Hall, 1990 and *Tutorial: Object-Oriented Computing*, Vol. 1, ed. G. Peterson, IEEE Press, 1987.

[8] Ken Kahn. The implementation of Uniform, a knowledge-representation programming language based upon equivalence of descriptions. In L. Steels and J. Campbell, editors, *Progress in Artificial Intelligence*. Ellis Horwood Limited Publishers, 1985. Earlier version appeared in ECAI-82, Orsay France, July 1982.

[7] Ken Kahn and Mats Carlsson. How to implement Prolog on a Lisp machine. In J. Campbell, editor, *Implementations of Prolog*. Ellis Horwood Ltd., 1984.

[6] Ken Kahn. Partial evaluation as an example of the relationships between programming methodology and artificial intelligence. *AI Magazine*, 5, Spring 1984. Also In *Artificial Intelligence Programming Environments*, ed. R. Hawley, Ellis Horwood, 1987 and In *Readings from the AI Magazine*, Volumes 1-5, ed. R. Engelmore, AAAI, 1988.

[5] Ken Kahn. A grammar kit in Prolog. In M. Yazdani, editor, *New Horizons in Educational Computing*. Ellis Horwood Ltd., 1984. Also In *Instructional Science* and *Proceedings of the AISB Easter Conference on AI and Education*, Exeter, England, April 1983.

[4] Ken Kahn. Intermission -- actors in Prolog. In S. Tarnlund and K. Clark, editors, *Logic Programming*. Academic Press, 1982. An earlier version was presented at the First Logic Programming Workshop, Hungary, 1980.

[3] Ken Kahn. Uniform -- a language based upon unification which unifies (much of) Lisp, Prolog, and Act 1. In *International Joint Conference on Artificial Intelligence*, August 1981. Also In *Logic Programming: Functions, Relations, and Equations,* eds. DeGroot, D. and Lindstrom, G., Prentice Hall, 1986.

[2] Ken Kahn and Tony Gorry. Mechanizing temporal knowledge. *Artificial Intelligence*, 9:87--108, August 1977.

[1] Ken Kahn. Three interactions between AI and education. In E. Elcock and D. Michie, editors, *Machine Intelligence 8: Machine Representations of Knowledge*. Ellis Horwood Ltd. and John Wylie & Sons, 1977.

Conference Proceedings:

[79] Ken Kahn. "Building Computer Models from Small Pieces", 2007 Summer Computer Simulation Conference, San Diego, CA, July 2007.

[78] Ken Kahn. Comparing Multi-Agent Models Composed from Micro-Behaviours, *Third International Model-to-Model Workshop*, Marseille, France, March 2007.

[76] Ken Kahn, "Time Travelling Animated Program Executions", *Proceedings of the Software Visualisation Conference,* 2006, Brighton, UK.

[75] Ken Kahn. A Program to Interpolate (and Extrapolate) Between Turtle Programs, *Proceedings of the Bridges Conference: Mathematical Connections in Art, Music, and Science*, August, 2006.

[73] Ken Kahn, Evgenia Sendova, Ana Isabel Sacristan, Richard Noss. Making Infinity Concrete by Programming Never-ending Processes, *Proceedings of the 7th International Conference on Technology and Mathematics Teaching*, Bristol, UK. July 2005.

[72] Morgado, Leonel; Morgado, Rosa; Cruz, Maria Gabriel; Kahn, Ken. Embedding Computer Activities into the Context of Preschools, in *Proceedings of Challenges 2005*, Paulo Dias & Cândido Varela de Freitas (eds.), ISBN 972-8746-13-05, pp. 471-478, Centro de Competências Nónio Séc. XXI, Universidade do Minho, Braga, Portugal. 2005.

[50] Ken Kahn. The Child-Engineering of Arithmetic in ToonTalk, *Proceedings of the Interaction Design and Children Conference*, College Park, Maryland, June 2004.

[49] Leonel Morgado, Maria Gabriel Cruz, and Ken Kahn. Taking Programming into Kindergartens. *EuroLogo Proceedings*, Portugal. August 2003.

[48] Tholander, J., Kahn, K., & Jansson, C.-G. (2002). Real Programming of an Adventure Game by an 8-year-old. *Proceedings of the International Conference of the Learning Sciences,* 2002, Seattle, WA.

[47] Ken Kahn. ToonTalk and Logo - Is ToonTalk a colleague, competitor, successor, sibling, or child of Logo? *Proceedings of the EuroLogo Conference*, August 2001.

[46] Ken Kahn. From Prolog and Zelda to ToonTalk*, Proceedings of the International Conference on Logic Programming*, edited by Danny De Schreye, MIT Press, 1999.

[45] Ken Kahn. A Computer Game to Teach Programming, *in Proceedings of the National Educational Computing Conference*, June 1999.

[44] Ken Kahn. Seeing Systolic Computations in a Video Game World, *Proceedings of the IEEE Conference on Visual Languages*, Bolder, Colorado, September 1996.

[43] Ken Kahn. Metaphor Design -- Case Study of an Animated Programming Environment, Proceedings of the Computer Game Developers Conference. April 1995.

[42] Ken Kahn. Programming as a Video Game. Proceedings of the Computer Game Developers Conference. April 1994.

[41] Kenneth M. Kahn. Concurrent constraint programs to parse and animate pictures of concurrent constraint programs. In *Proceedings of the International Conference on Fifth Generation Computer Systems*. ICOT, June 1992.

[40] Vijay A. Saraswat, Kenneth Kahn, and Jacob Levy. Janus--A step towards distributed constraint programming. In *Proceedings of the North American Logic Programming Conference*. MIT Press, October 1990.

[39] Kenneth M. Kahn and Vijay A. Saraswat. Complete visualizations of concurrent programs and their executions. In *Proceedings of the IEEE Visual Language Workshop*, October 1990.

[38] Kenneth Kahn and Vijay Saraswat. Actors as a special case of concurrent constraint programming. In *Proceedings of the Joint Conference on Object-Oriented Programming: Systems, Languages, and Applications and the European Conference on Object-Oriented Programming*. ACM Press, October 1990.

[37] K. Kahn and W. Kornfeld. Money as a concurrent logic program. In *Proceedings of the North American Conference on Logic Programming*. The MIT Press, 1989.

[36] Kenneth Kahn. Objects -- a fresh look. In Stephen Cook, editor, *Proceedings of the Third European Conference on Object-Oriented Programming*, pages 207--224. Cambridge University Press, July 1989.

[35] Vijay A. Saraswat, David Weinbaum, Ken Kahn, and Ehud Shapiro. Detecting stable properties of networks in concurrent logic programming languages. In *Proceedings of the Seventh Annual ACM Symposium on Principles of Distributed Computing (PODC 88)*, pages 210--222, August 1988.

[34] Eric Dean Tribble, Mark Miller, Ken Kahn, Daniel Bobrow, Curtis Abbott, and Ehud Shapiro. Channels: A generalization of streams. In *Proceedings of the Fourth International Conference on Logic Programming*, May 1987.

[33] Sanjay Mittal, Ken Kahn, and Daniel Bobrow. Virtual copies: At the boundary between classes and instances. In *Proceedings of the ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications*, September 1986.

[32] Ken Kahn, Eric Dean Tribble, Mark Miller, and Daniel Bobrow. Objects in concurrent logic programming languages. In *Proceedings of the ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications*, September 1986.

[31] Lars-Henrik Eriksson, Ken Kahn, and Manny Rayner. Incorporating mutable arrays into logic programming. Technical report, University of Uppsala, UPMAIL, July 1984. Also in *Proceedings of the Second International Logic Programming Conference*.

[30] Ken Kahn. A primitive for the control of logic programs. In *Proceedings of the IEEE International Symposium on Logic Programming*, February 1984.

[29] Ken Kahn. The compilation of Prolog programs without the use of a Prolog compiler. In *Proceedings of the Fifth Generation Computer Systems Conference,* 1984.

[28] Ken Kahn. A partial evaluator of Lisp written in Prolog. In *First International Logic Programming Conference*, September 1982.

[27] Ken Kahn. How to program a society. In *Artificial Intelligence and Simulation of Behavior Conference*, July 1980.

[26] Ken Kahn. Making aesthetic choices. In *International Joint Conference on Artificial Intelligence,* August 1979.

[25] Ken Kahn and Carl Hewitt. Dynamic graphics using quasi-parallelism. In *Proceedings of the ACM/SIGGRAPH Conference*, August 1978. Also AI Memo 480, MIT, June 1978.

[24] Ken Kahn. Ani: An example of computational creativity. In *Proceedings of AISB/GI Conference*, July 1978.

[23] Ken Kahn. An actor-based computer animation language. In S. Treu, editor, *Proceedings of the SIGGRAPH/ACM Workshop on User-Oriented Design of Interactive Graphics Systems*, pages 37--43, October 1976. Revision of: LOGO Working Paper 48, AI Working Paper 120, MIT, February 1976. Also published in: *Creative Computing*, Vol. 6, No. 11, Nov. 1980, pp. 75-84.

Technical reports and other publications:

[66] Ken Kahn and Ramana Rao. A circus of objects: The multi-faceted visualization of object-oriented designs. Technical report, Xerox PARC, 1992.

[65] Kenneth M. Kahn. Animation of Horn Clause proofs. Technical report, SSL, Xerox PARC, 1992.

[64] Vijay A. Saraswat, Kenneth M. Kahn, and Jacob Levy. Distributed constraint programming--the DC framework and Janus. Technical report, Xerox PARC, August 1989.

[63] K. Kahn and E. Shapiro. Logic programs with implicit state. Technical report, Weizmann Institute of Science, Rehovot, Israel, 1988.

[62] Ken Kahn and Mark Miller. Objects with state in Prologs with freeze. Technical report, ISL, Xerox PARC, 1987.

[61] Ken Kahn, Curtis Abbott, Daniel Bobrow, Mark Miller, and Eric Dean Tribble. Concurrent logic programming abstractions. Technical report, ISL, Xerox PARC, 1987.

[60] Susan Hirsh, Ken Kahn, and Mark Miller. Interming: Unifying positional and keyword notations. Technical report, Xerox PARC, 1987.

[59] Cara Holman, Alan Borning, Ken Kahn, and Mark Miller. Constraints and logic programming. Technical report, University of Washington, Dept. of Computer Science, December 1986.

[58] Mats Carlsson and Ken Kahn. LM-Prolog user manual. Technical Report 24, UPMAIL, Uppsala University, November 1983.

[57] Ken Kahn. Unique features of LM-Prolog. Technical report, Uppsala University UPMAIL, February 1983.

[56] Ken Kahn. Director guide. Technical Report 482B, MIT AI Lab, December 1979.

[55] Ken Kahn. Creation of computer animation from story descriptions. Technical Report 540, MIT AI Lab, August 1979.

[54] Ken Kahn and Henry Lieberman. Computer animation: Snow White's dream machine. *Technology Review*, 80:34--46, October/November 1977.

[53] Ken Kahn. A knowledge-based computer animation system. Technical report, MIT AI Lab, February 1976. LOGO Working Paper 47, AI Working Paper 119.

[52] Ken Kahn. Mechanization of temporal knowledge. Technical Report 155, MIT Project MAC, September 1975.

[51] Ken Kahn. A Logo natural language system. Technical report, MIT AI Lab, December 1975. LOGO Working Paper 46.

Films and plays:

[71] Kenneth M. Kahn. Towards visual concurrent constraint programming. Technical report, SSL, Xerox PARC, 1992. Contains a play written and directed by Ken Kahn, performed during San Francisco Bay Area ACM SIGCHI Seminar.

[70] Ken Kahn. Two nights on a computer. MIT AI Lab, March 1978. Computer animation, 4.5 minutes.

[69] Ken Kahn. Cinderella. MIT AI Lab, October 1978. Computer animation, 3 minutes.

[68] Ken Kahn. Mulberry bush. Carpenter Center for the Visual Arts, Harvard University, 1976. Hand-drawn animation, 3 minutes, honorable mention, First International Student Film Festival.

[67] Ken Kahn. Cons. Carpenter Center for the Visual Arts, Harvard University and MIT AI Lab, 1975. Computer and hand-drawn animation, 2 minutes.