

米澤明憲先生の研究教育活動

2011年2月23日

東京大学大学院情報理工学系研究科
コンピュータ科学専攻



目次

1	経歴	1
1.1	履歴	1
1.2	表彰	2
1.3	公職等（抜粋）	3
2	学会活動・招待講演等	4
2.1	学会活動等（抜粋）	4
2.2	招待講演（国際会議、抜粋）	5
3	論文・著作	7
3.1	著書	7
3.2	編書	7
3.3	英文論文誌論文	9
3.4	書籍の章	11
3.5	国際会議論文	12
3.6	和文論文誌論文	28
3.7	解説	32
3.8	国内会議論文	33
3.9	技術報告	46
3.10	その他	48
4	学位指導リスト	49
4.1	博士論文	49
4.2	修士論文	52
4.3	卒業論文	59
	米澤研究室関係者・OB 名簿	66
	論文: 私のソフトウェア研究	68
	論文: 並列オブジェクトによる大規模システムの実現	80

1 経歴

1.1 履歴

氏名 米澤 明憲 (よねざわ あきのり)
生年月日 昭和 22 年 6 月 17 日生
学位 Ph.D. in Computer Science,
Massachusetts Institute of Technology (昭和 53 年)
工学博士, 東京大学工学部計数工学科 (昭和 53 年)

学歴

昭和 45 年 3 月 東京大学工学部計数工学科卒業
昭和 47 年 3 月 東京大学大学院工学系研究科計数工学専門課程修士課程修了
昭和 53 年 2 月 米国マサチューセッツ工科大学電気及び計算機科学科
大学院博士課程修了

修士論文: “On a Proof Procedure of the First-Order Predicate Calculus”
Ph.D. thesis: “Specification and Verification Techniques for Parallel Programs
Based on Message Passing Semantics”

職歴

昭和 49 年 9 月 米国マサチューセッツ工科大学電気及び計算機科学科研究助手
昭和 53 年 1 月 東京工業大学理学部助手
昭和 58 年 3 月 東京工業大学理学部助教授
昭和 63 年 2 月 東京工業大学理学部教授
(昭和 63 年 6 月から 8 月まで米国メリーランド大学招聘教授)
平成元年 10 月 東京大学理学部教授
(平成 4 年 2 月から 3 月まで仏国パリ大学第 6 校招聘教授)
(平成 5 年 6 月から平成 9 年 3 月まで
北陸先端科学技術大学院大学教授併任)
(平成 11 年 4 月から平成 12 年 3 月まで
学術情報センター教授併任)
平成 12 年 4 月 東京大学大学院情報学環教授 (配置換)
(平成 12 年 11 月から平成 13 年 3 月まで
国立情報学研究所教授併任)
平成 15 年 4 月 東京大学大学院情報理工学系研究科教授 (配置換)
(平成 15 年 4 月から平成 16 年 3 月まで
国立情報学研究所教授併任)
平成 16 年 4 月 大学共同利用機関法人 情報・システム研究機構 監事
(平成 19 年 3 月まで)
平成 18 年 4 月 東京大学情報基盤センター長 (平成 22 年 3 月まで)
平成 18 年 9 月 独立行政法人産業技術総合研究所情報セキュリティ研究センター
副センター長 (平成 23 年 3 月まで)
平成 22 年 10 月 大学共同利用機関法人 情報・システム研究機構
ライフサイエンス統合データベースセンター長

1.2 表彰

昭和 56 年 5 月	情報処理学会英文誌論文賞 受賞 “Specifying Software Systems with High Internal Concurrency Based on Actor Formalism” (Akinori Yonezawa)
平成 4 年 11 月	大川情報通信財団出版賞 受賞 「モデルと表現」(米澤明憲, 柴山悦哉)
平成 6 年 10 月	IEEE Symposium on Visual Languages 論文賞 受賞 “A Framework for Constructing Animations via Declarative Mapping Rules” (Shin Takahashi, Ken Miyashita, Satoshi Matsuoka, Akinori Yonezawa)
平成 9 年 5 月	情報処理学会論文賞 受賞 「並列自己反映言語システムの部分計算によるコンパイル技法」 (増原英彦, 松岡聡, 米澤明憲)
平成 11 年 5 月	ACM (米国計算機学会) Fellow 顕彰理由「並列オブジェクト指向計算のパイオニア」
平成 12 年 9 月	IEEE (米国電気電子学会) The Joint Symposium ASA/MA2000 Best Paper Award “Bytecode Transformation for Portable Thread Migration in Java” (Takahiro Sakamoto, Tatsuro Sekiguchi, Akinori Yonezawa)
平成 14 年 5 月	情報処理学会論文賞 受賞 「排他的なメソッドの並行な呼び出しを融合する機構を持つ言語」 (大山恵弘, 田浦健次朗, 米澤明憲)
平成 15 年 6 月	日本ソフトウェア科学会論文賞 受賞 「移動計算のためのプログラミング言語 MobileML」 (橋本政朋, 米澤明憲)
平成 16 年 9 月	日本ソフトウェア科学会 フェロー
平成 19 年 9 月	FIT2007 第 6 回情報科学技術フォーラム論文賞 受賞 「OS 資源ビューの仮想化を用いた分散システムテストベッド」 (西川賀樹, 大山恵弘, 米澤明憲)
平成 20 年 7 月	AITO (国際オブジェクト技術協会) Dahl-Nygaard 賞 受賞
平成 20 年 9 月	日本ソフトウェア科学会 功労賞 受賞
平成 21 年 9 月	船井情報科学振興財団 FIT 船井業績賞 受賞
平成 21 年 11 月	紫綬褒章 受章
平成 22 年 5 月	情報処理学会 功績賞 受賞
平成 23 年 5 月	情報処理学会 フェロー (予定)

1.3 公職等（抜粋）

- 文部省 学術審議会専門委員（科学研究費分科会）（平成 2 年 2 月～平成 8 年 1 月）
- ドイツ国立情報科学技術研究所（GMD）科学顧問（平成 4 年 10 月～平成 8 年 9 月）
- 文部省 学術審議会専門委員（特定研究領域推進分科会）（平成 9 年 1 月～平成 10 年 1 月, 平成 12 年 11 月～平成 12 年 12 月）
- 文部省 科学技術専門委員（平成 10 年 3 月～平成 10 年 12 月）
- 日本学術振興会未来開拓事業「知能情報・高度情報」分野推進委員会委員長（平成 10 年～平成 14 年）
- 文部科学省特定領域研究「社会基盤としてのセキュアコンピューティングの実現方式の研究」（平成 12 年 10 月～平成 16 年 3 月）領域代表者
- 文部科学省 研究振興局情報科学技術委員会（平成 13 年 2 月～平成 21 年 1 月）
- 内閣府 総合規制改革会議委員（平成 13 年 5 月～平成 16 年 3 月）
- 文部科学省 科学技術・学術審議会専門委員（学術分科会）（平成 13 年 7 月～平成 16 年 1 月）
- 高度情報通信ネットワーク社会推進戦略本部 IT 関連規制改革専門調査委員（平成 13 年 12 月～平成 14 年 12 月）
- 大学共同利用機関法人 情報・システム研究機構 監事（平成 16 年 4 月～平成 19 年 3 月）
- 独立行政法人産業技術総合研究所情報セキュリティ研究センター 副センター長（平成 18 年 9 月～平成 23 年 3 月）
- 日本学術会議会員（情報学）（平成 20 年 10 月～）
- 大学共同利用機関法人 情報・システム研究機構 ライフサイエンス統合データベースセンター長（平成 22 年 10 月～）
- 科学技術振興機構 戦略的創造研究推進事業 研究領域「ポストペタスケール高性能計算に資するシステムソフトウェア技術の創出」研究総括（平成 22 年 10 月～）

2 学会活動・招待講演等

2.1 学会活動等（抜粋）

- 情報処理学会論文誌 編集委員（昭和54年6月～昭和58年5月）
- 情報処理学会欧文誌 編集委員（昭和58年6月～昭和63年5月）
- New Generation Computing (Springer-Verlag) 編集委員（昭和58年4月～平成4年3月）
- 日本ソフトウェア科学会理事（昭和59年4月～昭和63年および平成3年4月～平成6年3月）
- 日本ソフトウェア科学会「コンピュータソフトウェア」編集委員（昭和61年4月～平成元年3月）
- IEEE Computer 誌 編集委員（昭和63年2月～平成4年12月）
- ACM Conference on Object-Oriented Programming, Systems, Languages and Applications (ECOOP/OOPSLA '90) プログラム委員長
- International Series of Monographs in Computer Science, Oxford University Press, シリーズ編集委員（平成4年9月～）
- IEEE Parallel Distributed Technology 誌 編集委員（平成5年1月～平成8年12月）
- 論文誌 Theory and Practice of Object Systems, John-Wiley, Associate Editor（平成5年10月～平成8年9月）
- 日本ソフトウェア科学会理事長（平成7年4月～平成10年3月）
- ACM 論文誌 Transactions on Programming Languages and Systems (TOPLAS), Associate Editor (副編集長)（平成7年4月～平成8年6月）
- ACM プログラミング言語部会 (SIGPLAN) 幹事（平成7年7月～平成8年12月）

- その他国際会議プログラム委員多数

2.2 招待講演 (国際会議、抜粋)

- Object-Oriented Concurrent Programming and Its industrial Applications. International Conference on Theory and Practice of Software Development (TAPSOFT), Berlin, March 28, 1985.
- Object-Oriented Concurrent Computing. Department of Computer Science and Engineering, Technical University of Dresden, Dresden, East Germany, May 3, 1985.
- Object-Oriented Concurrent Programming - A Language ABCL -. AFCET Conference on Object-Oriented Languages, Paris, France, May 3, 1986.
- AI Parallelism and Programming. IFIP Congress, Dublin, Ireland, September 3, 1986.
- Object-Oriented Concurrent Computing. Fall 1987 Lecture Series, Department of Computer Science, University of Maryland, October 12, 1987.
- Reflection in an Object-Oriented Concurrent Language. School on Foundations of Object-Oriented Languages, Noorwijkerhout, Netherlands, May 29, 1990.
- Object-oriented Concurrent Computing. Five Lecture Series, Swedish Institute of Computer Science, June, 1992.
- Theory and Practice of Concurrent Object-Oriented Computing. International Conference on Theoretical Aspects of Computer Software (TACS '94), Sendai, Japan, April 20, 1994.
- Object-Based Models and Languages for Concurrent Systems. 1994 Workshop on Models and Languages for Coordination of Parallelism and Distribution, Bologna, Italy, July 5, 1994.
- Mobile Objects and their Implementations. International Symposium on Future Software Technology, Hangzhou, China, October 29, 1998.
- Distributed and Concurrent Objects Based on Linear Logic. Third International Conference on Formal Methods for Open Object-based Distributed Systems (FMOODS 1999), Florence, Italy, February 18, 1999.
- Message or Object? – Origin and Future of Concurrent/Mobile Objects –. The 7th International Workshop on Foundations of Object-Oriented Languages (FOOL 7), Boston, USA, January 22, 2000.
- Overview of the Japanese Inter-University Research Project on Software Security. 4th International Symposium on Theoretical Aspects of Computer Software (TACS '01), Sendai, Japan, October 30, 2001.

- An Overview of a Mext funded Inter-University Software Security Research, Berkeley, California, USA, August, 2002. An NSF Agenda Meeting on computer security.
- An Overview on a 3.5-Year Japanese Inter-University Research on Software Security. Workshop on New Approaches to Software Construction - WNASC 2004, Komaba, Tokyo, September 14, 2002.
- Concurrent Objects - Introspect and Prospect -. European Conference on Object Oriented Programming, Paphos, Cyprus, July 10, 2008.
- Modeling and Simulating Real/Virtual Worlds with Concurrent Objects. The 4th Franco-Japanese Computer Security Workshop, Tokyo, December 5, 2008.

3 論文・著作

3.1 著書

- 木村 泉, 米澤 明憲. 算法表現論. 岩波書店, May 1982. 294 pages.
- Akinori Yonezawa. *ABCL: An Object-Oriented Concurrent System*. MIT Press, February 1990. 343 pages.
- 米澤 明憲, 柴山 悦哉. モデルと表現. 岩波書店, April 1992. 336 pages.

3.2 編書

- Eiichi Goto, Koichi Furukawa, Reiji Nakajima, Ikuo Nakata, and Akinori Yonezawa, editors. *RIMS Symposium on Software Science and Engineering, 1982, Proceedings*, Vol. 147 of *Lecture Notes in Computer Science*. Springer, Kyoto, 1983. 232 pages.
- Akinori Yonezawa and Mario Tokoro, editors. *Object-Oriented Concurrent Programming*. MIT Press, April 1987. 282 pages.
- Gul Agha, Peter Wegner, and Akinori Yonezawa, editors. *Proceedings of the 1988 ACM SIGPLAN workshop on Object-Based Concurrent Programming*. ACM, San Diego, 1988. 214 pages.
- Mario Tokoro, Yuichiro Anzai, and Akinori Yonezawa, editors. *Concepts and Characteristics of Knowledge-Based Systems: selected and reviewed papers from the IFIP TC 10/WG10.1 workshop, November 9-12, 1987*. North-Holland, Mount Fuji, Japan, 1989. 511 pages.
- 長尾真, 稲垣康善, 辻井潤一, 中田育男, 石田晴久, 田中英彦, 所真理雄, 米澤明憲 (編). 岩波情報科学辞典. 岩波書店, May 1990. 1172 pages.
- Akinori Yonezawa and Takayasu Ito, editors. *Concurrency: Theory, Language, and Architecture, UK/Japan Workshop, September 25-27, 1989, Proceedings*, Vol. 491 of *Lecture Notes in Computer Science*. Springer, Oxford, UK, April 1991. 339 pages.
- 武市正人, 米澤明憲. (シリーズ編集)レクチャーノート・ソフトウェア科学. 近代科学社, 1992~.
- C. K. Yuen and Akinori Yonezawa, editors. *Parallel Programming Systems: Proceedings of a JSPS Seminar (Tokyo, Japan, 27-29 May 1992)*. World Scientific, September 1993. 253 pages.

- Gul Agha, Peter Wegner, and Akinori Yonezawa, editors. *Research Directions in Concurrent Object-Oriented Programming*. MIT Press, November 1993. 532 pages.
- Shojiro Nishio and Akinori Yonezawa, editors. *Object Technologies for Advanced Software, First JSSST International Symposium*, Vol. 742 of *Lecture Notes in Computer Science*. Springer, Kanazawa, November 1993. 543 pages.
- 米澤明憲, 松岡聡, 加藤和彦 (編). オブジェクト指向コンピューティング II, レクチャーノート/ソフトウェア学 (6), 日本ソフトウェア科学会 WOOC '93. 近代科学社, April 1994. 254 pages.
- Takayasu Ito and Akinori Yonezawa, editors. *Theory and Practice of Parallel Programming, TPPP '94 (November 1994)*, Vol. 907 of *Lecture Notes in Computer Science*. Springer, Sendai, 1995. 485 pages.
- Paolo Ciancarini, Oscar Nierstrasz, and Akinori Yonezawa, editors. *Object-Based Models and Languages for Concurrent Systems, ECOOP '94 Workshop on Models and Languages for Coordination of Parallelism and Distribution, July 5, 1994, Selected Papers*, Vol. 924 of *Lecture Notes in Computer Science*. Springer, Bologna, Italy, 1995. 193 pages.
- Jean-Pierre Briot, Jean-Marc Geib, and Akinori Yonezawa, editors. *Object-Based Parallel and Distributed Computing, France-Japan Workshop, OBPDC '95 (June 21-23), Selected Papers*, Vol. 1107 of *Lecture Notes in Computer Science*. Springer, Tokyo, 1996. 349 pages.
- Jean-Paul Bahsoun, Takanobu Baba, Jean-Pierre Briot, and Akinori Yonezawa, editors. *Object-Oriented Parallel and Distributed Programming*. HERMES Science Publications, Paris, January 2000. 329 pages.
- Akinori Yonezawa and Satoshi Matsuoka, editors. *The Third International Conference on Metalevel Architectures and Separation of Crosscutting Concerns, REFLECTION 2001, September 25-28*, Vol. 2192 of *Lecture Notes in Computer Science*. Springer, Kyoto, September 2001. 283 pages.
- Mitsuhiro Okada, Benjamin Pierce, Andre Scedrov, Hideyuki Tokuda, and Akinori Yonezawa, editors. *Software Security – Theories and Systems, Next-NSF-JSPS International Symposium, ISSS 2002 (November 8-10), Revised Papers*, Vol. 2609 of *Lecture Notes in Computer Science*. Springer, February 2003. 471 pages.

3.3 英文論文誌論文

- Akinori Yonezawa and Carl Hewitt. Modelling Distributed Systems. *Machine Intelligence*, Vol. 9, pp. 41–50, 1979.
- Akinori Yonezawa. Comments on Monitors and Path-Expressions. *Journal of Information Processing*, Vol. 1, No. 4, pp. 180–186, 1979.
- Akinori Yonezawa. Specifying Software Systems with High Internal Concurrency Based on Actor Formalism. *Journal of Information Processing*, Vol. 2, No. 4, pp. 208–218, 1980.
- Akinori Yonezawa. A Method for Synthesizing Data Retrieving Programs. *Journal of Information Processing*, Vol. 5, No. 2, pp. 94–101, 1982.
- Koichi Furukawa, Reiji Nakajima, and Akinori Yonezawa. Modularization and Abstraction in Logic Programming. *New Generation Computing*, Vol. 1, No. 2, pp. 169–177, 1983.
- Hiroshi Maruyama and Akinori Yonezawa. A Prolog-Based Natural Language Front-End System. *New Generation Computing*, Vol. 2, No. 1, pp. 91–99, 1984.
- Akinori Yonezawa, Hans Loeper, and Hans-Jörg Jäkel. The Rendezvous Concept - a Programming Tool for Parallel Processing. *Journal of Information Processing and Cybernetics*, Vol. 21, No. 9, pp. 429–440, September 1985.
- Satoshi Matsuoka, Shin Takahashi, Tomihisa Kamada, and Akinori Yonezawa. A General Framework for Bi-Directional Translation between Abstract and Pictorial Data. *ACM Transactions on Information Systems*, Vol. 10, No. 4, pp. 408–437, October 1992.
- Akinori Yonezawa, Satoshi Matsuoka, Masahiro Yasugi, and Kenjiro Taura. Implementing Concurrent Object-Oriented Languages on Multicomputers. *IEEE Parallel and Distributed Technology: Systems and Technology*, Vol. 1, No. 2, pp. 49–61, May 1993.
- Toru Sugimoto and Akinori Yonezawa. Multiple World Representation of Mental States for Dialogue Processing. *IEICE Transaction on Information and Systems*, Vol. E77-D, No. 2, pp. 192–208, February 1994.
- Naoki Kobayashi and Akinori Yonezawa. Asynchronous Communication Model Based on Linear Logic. *Formal Aspects of Computing*, Vol. 7, No. 2, pp. 113–149, 1995.
- Naoki Kobayashi and Akinori Yonezawa. Towards Foundations of Concurrent Object-Oriented Programming-Types and Language Design. *Theory and Practice of Object Systems*, Vol. 1, No. 4, pp. 243–268, 1995.

- Akihiro Nakaya, Kenji Yamamoto, and Akinori Yonezawa. RNA Secondary Structure Prediction Using Highly Parallel Computers. *Computer Applications in the Biosciences*, Vol. 11, No. 6, pp. 685–692, December 1995.
- Satoshi Matsuoka, Masahiro Yasugi, Kenjiro Taura, Tomio Kamada, and Akinori Yonezawa. Compiling and Managing Concurrent Objects for Efficient Execution on High-Performance MPPs. In Lubomir Bic, Alexandru Nicolau, and Mitsuhsa Sato, editors, *Parallel Language and Compiler Research in Japan*, pp. 91–125. Kluwer Academic Publishers, 1995.
- Kenichi Asai, Satoshi Matsuoka, and Akinori Yonezawa. Duplication and Partial Evaluation For a Better Understanding of Reflective Languages. *Lisp and Symbolic Computation*, Vol. 9, No. 2/3, pp. 203–241, May 1996.
- Akihiro Nakaya, Kenjiro Taura, Kenji Yamamoto, and Akinori Yonezawa. Visualization of RNA Secondary Structures Using Highly Parallel Computers. *Computer Applications in the Biosciences*, Vol. 12, No. 3, pp. 205–211, June 1996.
- Akihiro Nakaya, Akinori Yonezawa, and Kenji Yamamoto. Classification of RNA Secondary Structures Using the Techniques of Cluster Analysis. *Journal of Theoretical Biology*, Vol. 183, No. 1, pp. 105–117, November 1996.
- Naoki Kobayashi, Toshihiro Shimizu, and Akinori Yonezawa. Distributed Concurrent Linear Logic Programming. *Theoretical Computer Science (Linear Logic Special Issues)*, Elsevier Science, Vol. 227, No. 1-2, pp. 185–220, September 1999.
- Yutaka Oiwa, Kenjiro Taura, and Akinori Yonezawa. Extending Java Virtual Machine with Integer-Reference Conversion. *Concurrency: Practice and Experience*, Vol. 12, No. 6, pp. 407–422, May 2000.
- Hidehiko Masuhara and Akinori Yonezawa. A Portable Approach to Dynamic Optimization in Run-time Specialization. *New Generation Computing*, Vol. 20, No. 1, pp. 101–124, November 2001.
- Etsuya Shibayama and Akinori Yonezawa. Secure Software Infrastructure in the Internet Age. *New Generation Computing*, Vol. 21, No. 2, pp. 87–106, February 2003.
- Kenji Kaneda, Kenjiro Taura, and Akinori Yonezawa. Virtual private grid: a command shell for utilizing hundreds of machines efficiently. *Future Generation Computer Systems*, Vol. 19, No. 4, pp. 563–573, May 2003.

3.4 書籍の章

- 柴山悦哉, 松田裕幸, 米澤明憲. 並列オブジェクト指向言語 ABCL によるプログラミング. オブジェクト指向 解説と WOOC'85 からの論文, pp. 57–82. 共立出版, December 1985.
- Etsuya Shibayama and Akinori Yonezawa. Distributed Computing in ABCL/1. In Akinori Yonezawa and Mario Tokoro, editors, *Object-Oriented Concurrent Programming*, pp. 91–128. MIT Press, 1987.
- Akinori Yonezawa et al. Modelling and Programming in an Object-Oriented Concurrent Language ABCL/1. In Akinori Yonezawa and Mario Tokoro, editors, *Object-Oriented Concurrent Programming*, pp. 55–90. MIT Press, 1987.
- Akinori Yonezawa, Jean-Pierre Briot, and Etsuya Shibayama. Object-Oriented Concurrent Programming in ABCL/1. In Alan H. Bond and Les Gasser, editors, *Readings in Distributed Artificial Intelligence*, pp. 434–444. Morgan Kaufman Publishers, 1988.
- Satoshi Matsuoka and Akinori Yonezawa. Analysis of Inheritance Anomaly in Object-Oriented Concurrent Programming Languages. In Gul Agha, Peter Wegner, and Akinori Yonezawa, editors, *Research Directions in Concurrent Object-Oriented Programming*, chapter 4, pp. 107–150. MIT Press, 1993.
- Akinori Yonezawa and Ichiro Osawa. Object-Oriented Parallel Parsing for Context-Free Grammars. In Geert Adriaens and Udo Hahn, editors, *Parallel Natural Language Processing*, chapter 4, pp. 188–210. Ablex Publishing Corporation, 1994.
- Akinori Yonezawa, Jean-Pierre Briot, and Etsuya Shibayama. Object-Oriented Concurrent Programming in ABCL/1. In David B. Skillicorn and Domenico Talia, editors, *Programming Languages for Parallel Processing*, pp. 158–168. IEEE Computer Society Press, 1995.
- Hidehiko Masuhara and Akinori Yonezawa. Reflection in Concurrent Object-Oriented Languages. In Howard Bowman and John Derrick, editors, *Formal Methods for Distributed Processing: A Survey of Object-Oriented Approaches*, pp. 305–325. Cambridge University Press, 2001.

3.5 国際会議論文

- Akinori Yonezawa and Carl Hewitt. Modelling Distributed Systems. In R. Reddy, editor, *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, pp. 370–376, Cambridge, MA, August 1977. William Kaufmann. Also in *Machine Intelligence 9*, Ellis Horwood Ltd., Chichester, Sussex, 1978.
- Akinori Yonezawa. A Specification Technique for Abstract Data Types with Parallelism. In *International Conference on Mathematical Studies of Information Processing*, Kyoto, Japan, August 1978. Also available as Research Report C-17, Department of Information Science, Tokyo Institute of Technology, April, 1978.
- Akinori Yonezawa. A Formal Specification Technique for Abstract Data Types with Parallelism. In *Mathematical Studies of Information Processing, Proceedings of the International Conference, August 23-26, 1978*, Vol. 75 of *Lecture Notes in Computer Science*, pp. 127–150, Kyoto, Japan, 1979. Springer.
- Akinori Yonezawa and Yoshihiro Matsumoto. Object Oriented Concurrent Programming and Industrial Software Production. In *Mathematical Foundations of Software Development, Proceedings of the International Joint Conference on Theory and Practice of Software Development (TAPSOFT), Volume 2: Colloquium on Software Engineering (CSE)*, Vol. 186 of *Lecture Notes in Computer Science*, pp. 395–409, Berlin, Germany, March 1985. Springer.
- Akinori Yonezawa. AI Parallelism and Programming. In H. J. Kugler, editor, *Information Processing 86, IFIP Congress*, pp. 111–113. North-Holland, 1986.
- Akinori Yonezawa, Hiroyuki Matsuda, and Etsuya Shibayama. An Approach to Object-Oriented Concurrent Programming – A Language ABCL –. In *Proceedings of AFCET Conference on Object-Oriented Languages*, pp. 125–134, Paris, 1986.
- Jean-Pierre Briot and Akinori Yonezawa. Inheritance mechanisms in distributed object-oriented languages. In *Conference on Software Science and Engineering (SSE'86)*, RIMS, Kyoto University, Japan, September 1986.
- Akinori Yonezawa, Jean-Pierre Briot, and Etsuya Shibayama. Object-Oriented Concurrent Programming in ABCL/1. In *Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '86)*, Portland, Oregon, November 1986. Also in *SIGPLAN Notices* 21(11):258–268.
- Jean-Pierre Briot and Akinori Yonezawa. Inheritance and Synchronization in Concurrent OOP. In *Proceedings of European Conference on Object-Oriented Programming (ECOOP '87)*, Vol. 276 of *Lecture Notes in Computer Science*, pp. 32–40, Paris, France, June 1987. Springer.

- Yasuaki Honda and Akinori Yonezawa. Debugging Concurrent Systems Based on Object Groups. In *Proceedings of European Conference on Object-Oriented Programming (ECOOP '88)*, Vol. 322 of *Lecture Notes in Computer Science*, pp. 267–282, Oslo, Norway, August 1988. Springer.
- Akinori Yonezawa and Ichiro Ohsawa. Object-Oriented Parallel Parsing for Context-Free Grammars. In *Proceedings of the 12th International Conference on Computational Linguistics*, pp. 773–778, Budapest, Hungary, August 1988.
- Garrison W. Cottrell, Pradip Dey, Joachim Diederich, Peter A. Reich, Lokendra Shastri, and Akinori Yonezawa. Parallel processing in computational linguistics. In *Proceedings of the 12th International Conference on Computational Linguistics*, pp. 595–598, Budapest, Hungary, August 1988.
- Takuo Watanabe and Akinori Yonezawa. Reflection in an Object-Oriented Concurrent Language. In *Proceedings of Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '88)*, pp. 306–315, San Diego, CA, September 1988. Also in SIGPLAN Notices 23(11), November 1988.
- Takuo Watanabe Akinori Yonezawa. An Introduction to Object-Based Reflective Concurrent Computation. In Gul Agha, Peter Wegner, and Akinori Yonezawa, editors, *Proceedings of the 1988 ACM SIGPLAN Workshop on Object-Based Concurrent Programming*, pp. 50–54, San Diego, 1989. Also in SIGPLAN Notices 24(4).
- Takuo Watanabe and Akinori Yonezawa. Reflective Computation in Object-Oriented Concurrent Systems and its Applications. In *Proceedings of the Fifth International Workshop on Software Specification and Design (IWSSD '89)*, pp. 56–58, Pittsburgh, USA, 1989.
- Akinori Yonezawa. A Reflective Object Oriented Concurrent Language ABCL/R. In *Proceedings of US/Japan Workshop on Parallel Lisp: Languages and Systems, June 1989*, Vol. 441 of *Lecture Notes in Computer Science*, pp. 254–256, Sendai, Japan, 1990. Springer.
- Etsuya Shibayama and Akinori Yonezawa. Optimistic and Pessimistic Synchronization in Distributed Computing. In *Proceedings of US/Japan Workshop on Parallel Lisp: Languages and Systems, June 5-8, 1989*, Vol. 441 of *Lecture Notes in Computer Science*, pp. 257–260, Sendai, Japan, 1990. Springer.
- Satoshi Matsuoka and Akinori Yonezawa. Metalevel solution to inheritance anomaly in concurrent object-oriented languages. In *ECOOP/OOPSLA '90 Workshop on Reflection and Metalevel Architectures in Object-Oriented Programming*, Ottawa, Canada, October 1990.

- Yuuji Ichisugi and Akinori Yonezawa. Exception Handling and Real Time Features in an Object-Oriented Concurrent Language. In *Concurrency: Theory, Language, and Architecture, UK/Japan Workshop, September 25-27, 1989, Proceedings*, Vol. 491 of *Lecture Notes in Computer Science*, pp. 92–109, Oxford, UK, 1991. Springer.
- Yuuji Ichisugi, Satoshi Matsuoka, Takuo Watanabe, and Akinori Yonezawa. An Object-Oriented Concurrent Reflective Architecture for Distributed Computing Environments (Extended Abstract). In *Proceedings of 29th Annual Allerton Conference on Communication, Control and Computing*, Allerton Illinois, 1991.
- Takuo Watanabe and Akinori Yonezawa. An Actor-Based Metalevel Architecture for Group-Wide Reflection. In *Proceedings of Foundations of Object-Oriented Languages, REX School/Workshop*, Vol. 489 of *Lecture Notes in Computer Science*, pp. 405–425, Noordwijkerhout, The Netherlands, May-June 1991. Springer.
- Naoki Kobayashi, Satoshi Matsuoka, and Akinori Yonezawa. Control in Parallel Constraint Logic Programming. In *Proceedings of Logic Programming Conference '91*, Lecture Notes in Artificial Intelligence, Tokyo, Japan, July 1991. Springer.
- Satoshi Matsuoka, Takuo Watanabe, and Akinori Yonezawa. Hybrid Group Reflective Architecture for Object-Oriented Concurrent Reflective Programming. In *Proceedings of European Conference on Object-Oriented Programming (ECOOP '91)*, Vol. 512 of *Lecture Notes in Computer Science*, pp. 231–250, Geneva, Switzerland, July 1991. Springer.
- Satoshi Matsuoka, Shin'ichi Furuso, and Akinori Yonezawa. A Fast Parallel Conservative Garbage Collector for Concurrent Object-Oriented Systems. In *Proceedings of IEEE International Workshop on Object Orientation in Operating Systems (I-WOOS '91)*, pp. 87–93, Palo Alto, CA, October 1991.
- Masahiro Yasugi and Akinori Yonezawa. Towards User (Application) Language-Level Garbage Collection in Object-Oriented Concurrent Languages. In *Proceedings of the OOPSLA '91 Workshop on Reflection and Metalevel Architectures in Object-Oriented Programming*, Phoenix, USA, October 1991.
- Ken Wakita and Akinori Yonezawa. Linguistic Supports for Development of Distributed Organizational Information Systems in Object-Oriented Concurrent Computation Frameworks. In *Proceedings of ACM Conference on Organizational Computing Systems (COCS 1991)*, pp. 185–198, Atlanta, November 1991.
- Shin Takahashi, Satoshi Matsuoka, Akinori Yonezawa, and Tomihisa Kamada. A General Framework for Bi-Directional Translation between Abstract and Pictorial Data. In *Proceedings of the Fourth ACM Symposium on User Interface Software and Technology (UIST '91)*, pp. 165–174, Hilton Head, South Carolina, USA, November 1991.

- Satoshi Matsuoka, Takuo Watanabe, Yuuji Ichisugi, and Akinori Yonezawa. Object-Oriented Concurrent Reflective Architectures. In *Proceedings of Object-Based Concurrent Computing, ECOOP '91 Workshop, July 15-16, 1991*, Vol. 612 of *Lecture Notes in Computer Science*, pp. 211–226, Geneva, Switzerland, 1992. Springer.
- Pierre America, Robin Milner, Oscar Nierstrasz, Mario Tokoro, and Akinori Yonezawa. What Is An Object? (panel). In *Proceedings of ECOOP '91 Workshop, Object-Based Concurrent Computing, July, 1991*, Vol. 612 of *Lecture Notes in Computer Science*, pp. 257–264, Geneva, Switzerland, 1992. Springer.
- Masahiro Yasugi, Satoshi Matsuoka, and Akinori Yonezawa. ABCL/onEM-4: A New Software/Hardware Architecture for Object-Oriented Concurrent Computing on an Extended Dataflow Supercomputer. In *Proceedings of the 6th International Conference on Supercomputing (ICS '92)*, pp. 93–103, Washington D.C., July 1992.
- Hidehiko Masuhara, Satoshi Matsuoka, Takuo Watanabe, and Akinori Yonezawa. Object-Oriented Concurrent Reflective Languages can be Implemented Efficiently. In *Proceedings of Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '92)*, pp. 127–144, Vancouver, Canada, October 1992. Also in SIGPLAN Notices 27(10).
- Yuuji Ichisugi, Satoshi Matsuoka, and Akinori Yonezawa. RbCl: A Reflective Concurrent Language without a Run-Time Kernel. In *Proceedings of IMSA '92 International Workshop on Reflection and Metalevel Architectures*, Tokyo, Japan, November 1992.
- Ken Miyashita, Satoshi Matsuoka, Shin Takahashi, Akinori Yonezawa, and Tomihisa Kamada. Declarative Programming of Graphical Interfaces by Visual Examples. In *Proceedings of the Fifth ACM Symposium on User Interface Software and Technology (UIST '92)*, pp. 107–116, Monterey, CA, November 1992.
- Akinori Yonezawa, Satoshi Matsuoka, Masahiro Yasugi, and Kenjiro Taura. Efficient Implementations of Concurrent Object-Oriented Languages on Multicomputers (abstract). In *Proceedings of a JSPS Seminar: Parallel Programming Systems*, pp. 50–52, 1993.
- Kenjiro Taura, Satoshi Matsuoka, and Akinori Yonezawa. An Efficient Implementation Scheme of Concurrent Object-Oriented Languages on Stock Multicomputers. In *Proceedings of US/Japan Workshop on Parallel and Symbolic Computing: Languages, Systems, and Applications, October, 1992*, Vol. 748 of *Lecture Notes in Computer Science*, pp. 402–405, Cambridge, MA, USA, 1993. Springer.
- Naoki Kobayashi and Akinori Yonezawa. Asynchronous Communication Model Based on Linear Logic. In *Proceedings of US/Japan Workshop on Parallel and*

Symbolic Computing: Languages, Systems, and Applications, October, 1992, Vol. 748 of *Lecture Notes in Computer Science*, pp. 331–336, Cambridge, MA, USA, 1993. Springer.

- Kenjiro Taura, Satoshi Matsuoka, and Akinori Yonezawa. An Efficient Implementation Scheme of Concurrent Object-Oriented Language on Stock Multicomputers. In *Proceedings of the Fourth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP '93), May 1993*, pp. 218–228, San Diego, May 1993. Also in SIGPLAN Notices 28(7), July, 1993.
- Takashi Miyata, Kôiti Hasida, and Akinori Yonezawa. Plan Inferences in Dialogue under Dynamical Constraint Programming. In *Proceedings of the Fourth International Workshop on Natural Language Understanding and Logic Programming (NLULP,4)*, pp. 129–145, Nara, Japan, September–October 1993.
- Kenichi Asai, Satoshi Matsuoka, and Akinori Yonezawa. Model Checking of Control-Finite CSP Programs. In *Proceedings of the 26th Hawaii International Conference on Systems Science*, Vol. 2, pp. 174–183, 1993.
- Naoki Kobayashi and Akinori Yonezawa. ACL – A Concurrent Linear Logic Programming Paradigm. In *Proceedings of the 1993 International Logic Programming Symposium*, pp. 279–294, Vancouver, Canada, October 1993. MIT Press.
- Hidehiko Masuhara, Satoshi Matsuoka, and Akinori Yonezawa. Designing an object-oriented reflective language for massively-parallel processors. In *Proceedings of OOPSLA '93 Workshop on Object-Oriented Reflection and Metalevel Architectures*, Washington D.C., USA, October 1993.
- Kenichi Asai, Satoshi Matsuoka, and Akinori Yonezawa. Duplication and Partial Evaluation to Implement Reflective Languages. In *Proceedings of OOPSLA '93 Workshop on Reflection and Metalevel Architectures*, Washington D.C., USA, October 1993.
- Satoshi Matsuoka, Kenjiro Taura, and Akinori Yonezawa. Highly Efficient and Encapsulated Re-use of Synchronization Code in Concurrent Object-Oriented Languages. In *Proceedings of the 8th Annual Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '93)*, pp. 109–126, Washington D.C., USA, September–October 1993. Also in SIGPLAN Notices 28(10).
- Mehmet Aksit, Ken Wakita, Jan Bosch, Lodewijk Bergmans, and Akinori Yonezawa. Abstracting Object Interactions Using Composition Filters. In *Proceedings of ECOOP '93 Workshop, Object-Based Distributed Computing, July, 1993*, Vol. 791 of *Lecture Notes in Computer Science*, pp. 152–184, Kaiserslautern, Germany, 1994. Springer.

- Tatsuou Sekiguchi and Akinori Yonezawa. A Complete Type Inference System for Subtyped Recursive Types. In *Proceedings of International Conference on Theoretical Aspects of Computer Software (TACS '94)*, Vol. 789 of *Lecture Notes in Computer Science*, pp. 667–686, Sendai, Japan, April 1994. Springer.
- Kenjiro Taura, Satoshi Matsuoka, and Akinori Yonezawa. ABCL/f: A Future-Based Polymorphic Typed Concurrent Object-Oriented Language – Its Design and Implementation –. In *Proceedings of the DIMACS workshop on Specification of Parallel Algorithms*, Vol. 18 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pp. 275–291, Princeton, New Jersey, May 1994.
- Hiroshi Hosobe, Ken Miyashita, Shin Takahashi, Satoshi Matsuoka, and Akinori Yonezawa. Locally Simultaneous Constraint Satisfaction. In *Proceedings of Second International Workshop on Principles and Practice of Constraint Programming (PPCP '94)*, Vol. 874 of *Lecture Notes in Computer Science*, pp. 51–62, Rosario, Orcas Island, Washington, USA, May 1994. Springer.
- Masahiro Yasugi, Satoshi Matsuoka, and Akinori Yonezawa. The Plan-Do Style Compilation Technique for Eager Data Transfer in Thread-Based Execution. In *Proceedings of the IFIP WG10.3 International Conference on Parallel Architectures and Compilation Techniques (PACT '94)*, pp. 57–66, Montréal, Canada, August 1994.
- Kazuhiro Konno, Masaaki Nagatsuka, Naoki Kobayashi, Satoshi Matsuoka, and Akinori Yonezawa. PARCS: An MPP-Oriented CLP Language. In *Proceedings of the First International Symposium on Parallel Symbolic Computation (PASCO '94)*, pp. 254–263, Linz, Austria, September 1994. World Scientific.
- Shin Takahashi, Ken Miyashita, Satoshi Matsuoka, and Akinori Yonezawa. A Framework for Constructing Animations via Declarative Mapping Rules. In *Proceedings of IEEE Symposium on Visual Languages*, pp. 314–322, St. Louis, Missouri, October 1994.
- Naoki Kobayashi and Akinori Yonezawa. Type-Theoretic Foundations for Concurrent Object-Oriented Programming. In *Proceedings of the 9th Annual Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '94)*, pp. 31–45, Portland, Oregon, October 1994. Also in SIGPLAN Notices 29(10).
- Ken Miyashita, Satoshi Matsuoka, Shin Takahashi, and Akinori Yonezawa. Interactive Generation of Graphical User Interfaces by Multiple Visual Examples. In *Proceedings of the 7th ACM Symposium on User Interface Software and Technology '94 (UIST'94)*, pp. 85–94, Marina del Rey, CA, USA, November 1994.
- Tomio Kamada, Satoshi Matsuoka, and Akinori Yonezawa. Efficient Parallel Global Garbage Collection on Massively Parallel Computers. In *Proceedings of*

the 1994 ACM/IEEE Conference on Supercomputing (Supercomputing '94), pp. 79–88, Washington D.C., November 1994.

- Tomio Kamada, Satoshi Matsuoka, and Akinori Yonezawa. An Algorithm for Efficient Global Garbage Collection on Massively Parallel Computers. In *Proceedings of International Workshop on Theory and Practice of Parallel Programming (TPPP '94), November, 1994*, Vol. 907 of *Lecture Notes in Computer Science*, pp. 346–355, Sendai, Japan, 1995. Springer.
- Naoki Kobayashi and Akinori Yonezawa. Higher-Order Concurrent Linear Logic Programming. In *Proceedings of International Workshop on Theory and Practice of Parallel Programming (TPPP '94), November, 1994*, Vol. 907 of *Lecture Notes in Computer Science*, pp. 137–166, Sendai, Japan, 1995. Springer.
- Kenjiro Taura, Satoshi Matsuoka, and Akinori Yonezawa. *StackThreads*: An Abstract Machine for Scheduling Fine-Grain Threads on Stock CPUs. In *Proceedings of International Workshop on Theory and Practice of Parallel Programming (TPPP '94), November, 1994*, Vol. 907 of *Lecture Notes in Computer Science*, pp. 121–136, Sendai, Japan, 1995. Springer.
- Naoki Kobayashi, Motoki Nakade, and Akinori Yonezawa. Static Analysis of Communication for Asynchronous Concurrent Programming Languages. In *Proceedings of Second International Static Analysis Symposium (SAS'95)*, Vol. 983 of *Lecture Notes in Computer Science*, pp. 225–242, Glasgow, UK, September 1995. Springer.
- Shin Takahashi, Satoshi Matsuoka, Ken Miyashita, Hiroshi Hosobe, Akinori Yonezawa, and Tomihisa Kamada. A Constraint-Based Approach for Visualization and Animation. In *Proceedings of the International Workshop on Constraints for Graphics and Visualization (in conjunction with CP '95)*, pp. 103–117, Cassis, France, September 1995.
- Hidehiko Masuhara, Satoshi Matsuoka, Kenichi Asai, and Akinori Yonezawa. Compiling Away the Meta-Level in Object-Oriented Concurrent Reflective Languages Using Partial Evaluation. In *Proceedings of the 10th Annual Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '95)*, pp. 300–315, Austin, Texas, USA, October 1995. Also in SIGPLAN Notices 30(10).
- Hidehiko Masuhara, Satoshi Matsuoka, Kenichi Asai, and Akinori Yonezawa. Efficient implementation technique for object-oriented concurrent reflective languages using partial evaluation. In *Proceedings of International Workshop on New Models for Software Architecture (IMSA '95)*, Tokyo, Japan, October 1995.
- Kenjiro Taura and Akinori Yonezawa. Schematic: A Concurrent Object-Oriented Extension to Scheme. In *Proceedings of Workshop on Object-Based Parallel and*

Distributed Computing (OBPDC '95), June, 1995, Vol. 1107 of *Lecture Notes in Computer Science*, pp. 59–82, Tokyo, Japan, 1996. Springer.

- Tomio Kamada and Akinori Yonezawa. A Debugging Scheme for Fine-Grain Threads on Massively Parallel Processors with a Small Amount of Log Information - Replay and Race Detection. In *Proceedings of International Workshop on Parallel Symbolic Languages and Systems (PSLS '95)*, October, 1995, Vol. 1068 of *Lecture Notes in Computer Science*, pp. 108–127, Beaune, France, 1996. Springer.
- Naohito Sato, Satoshi Matsuoka, and Akinori Yonezawa. Hierarchical Collections: An Efficient Scheme to Build an Obeject-Oriented Distributed Class Library for Massively Parallel Computation. In *Proceedings of the Second JSSST International Symposium on Object Technologies for Advanced Software (ISOTAS '96)*, Vol. 1049 of *Lecture Notes in Computer Science*, pp. 96–117, Kanazawa, Japan, March 1996. Springer.
- Hidehiko Masuhara, Satoshi Matsuoka, and Akinori Yonezawa. Implementing Parallel Language Constructs Using a Reflective Object-Oriented Language. In *Proceedings of Reflection '96*, pp. 79–91, San Francisco, April 1996.
- Hiroshi Hosobe, Satoshi Matsuoka, and Akinori Yonezawa. Generalized Local Propagation: A Framework for Solving Constraint Hierarchies. In *Proceedings of the Second International Conference on Principles and Practice of Constraint Programming (CP '96)*, Vol. 1118 of *Lecture Notes in Computer Science*, pp. 237–251, Cambridge, MA, USA, August 1996. Springer.
- Haruo Hosoya, Naoki Kobayashi, and Akinori Yonezawa. Partial Evaluation Scheme for Concurrent Languages and Its Correctness. In *Proceedings of the Second International Euro-Par Conference (Euro-Par '96)*, Vol. 1123 of *Lecture Notes in Computer Science*, pp. 625–632, Lyon, France, August 1996. Springer.
- Kenichi Asai, Hidehiko Masuhara, and Akinori Yonezawa. Partial Evaluation of Call-by-Value lambda-Calculus with Side-Effects. In *Proceedings ACM Conference on Partial Evaluation and Semantics-Based Program Manipulation (PEPM '97)*, pp. 12–21, Amsterdam, June 1997.
- Kenjiro Taura and Akinori Yonezawa. Fine-grain Multithreading with Minimal Compiler Support - A Cost Effective Approach to Implementing Efficient Multithreading Languages. In *Proceedings of the 1997 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '97)*, pp. 320–333, Las Vegas, June 1997.
- Kenjiro Taura and Akinori Yonezawa. An Effective Garbage Collection Strategy for Parallel Programming Languages on Large Scale Distributed-Memory Machines. In *Proceedings of ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP '97)*, pp. 264–275, Las Vegas, June 1997.

- Tatsurou Sekiguchi and Akinori Yonezawa. A Calculus with Code Mobility. In *Proceedings of Second IFIP International Conference on Formal Methods for Open Object-based Distributed Systems (FMOODS 1997)*, pp. 21–36, Canterbury, UK, July 1997.
- Naohito Sato, Satoshi Matsuoka, Jean-Marc Jezequel, and Akinori Yonezawa. A Methodology for Specifying Data Distribution using only Standard Object-Oriented Features. In *Proceedings of the 11th ACM International Conference on Supercomputing (ICS '97)*, pp. 116–123, Vienna, July 1997.
- Yoshihiro Oyama, Kenjiro Taura, and Akinori Yonezawa. An Efficient Compilation Framework for Languages Based on a Concurrent Process Calculus. In *Proceedings of Euro-Par '97, Object-Oriented Programming*, Vol. 1300 of *Lecture Notes in Computer Science*, pp. 546–553, Passau, Germany, August 1997. Springer.
- Hidehiko Masuhara and Akinori Yonezawa. Reasoning-conscious Meta-object Design of a Reflective Concurrent Language. In *Proceedings of International Symposium on Biologically Inspired Computation (IMSA '97)*, pp. 42–56, Tsukuba, Japan, October 1997. ETL/IPA.
- Toshio Endo, Kenjiro Taura, and Akinori Yonezawa. A Scalable Mark-Sweep Garbage Collector on Large-Scale Shared-Memory Machines. In *Proceedings of ACM/IEEE High Performance Computing and Networking (SC '97)*, San Jose, CA, USA, November 1997.
- Toshiyuki Takahashi, Yutaka Ishikawa, Mitsuhisa Sato, and Akinori Yonezawa. A Compile-Time Meta-Level Architecture Supporting Class Specific Optimization. In *Proceedings of International Symposium on Computing in Object-Oriented Parallel Environments (ISCOPE 1997)*, Vol. 1343 of *Lecture Notes in Computer Science*, pp. 89–96, Marina del Rey, CA, USA, December 1997. Springer.
- Haruo Hosoya and Akinori Yonezawa. Garbage Collection via Dynamic Type Inference — A Formal Treatment —. In *Proceedings of the Second Workshop on Types in Compilation (TIC '98)*, Vol. 1473 of *Lecture Notes in Computer Science*, pp. 215–239, Kyoto, Japan, March 1998. Springer.
- Hidehiko Masuhara and Akinori Yonezawa. A Reflective Approach to Support Software Evolution. In *Proceedings of International Workshop on Principles of Software Evolution (IWPSE'98)*, pp. 135–139, Kyoto, Japan, April 1998.
- Hirotaka Yamamoto, Kenjiro Taura, and Akinori Yonezawa. Comparing Reference Counting and Global Mark-and-Sweep on Parallel Computers. In *Proceedings of 4th International Workshop on Languages, Compilers, and Run-Time Systems for Scalable Computers (LCR '98)*, Vol. 1511 of *Lecture Notes in Computer Science*, pp. 205–218, Pittsburgh, USA, May 1998. Springer.

- Hidehiko Masuhara and Akinori Yonezawa. Design and Partial Evaluation of Meta-Objects for a Concurrent Reflective Language. In *Proceedings of 12th European Conference on Object-Oriented Programming (ECOOP '98)*, Vol. 1445 of *Lecture Notes in Computer Science*, pp. 418–439, Brussels, Belgium, July 1998.
- Yoshihiro Oyama, Kenjiro Taura, and Akinori Yonezawa. An Implementation and Performance Evaluation of Language with Fine-Grain Thread Creation on Shared Memory Parallel Computer. In *Proceedings of 1998 International Conference on Parallel and Distributed Computing and Systems (PDCS '98)*, pp. 672–675, Las Vegas, USA, October 1998.
- Yuuya Sugita, Hidehiko Masuhara, Ken'ichi Harada, and Akinori Yonezawa. On-the-fly Specialization of Reflective Programs Using Dynamic Code Generation Techniques. In *Proceedings of OOPSLA '98 workshop on Reflective Programming in C++ and Java*, Vancouver, Canada, October 1998.
- Tatsuro Sekiguchi, Hidehiko Masuhara, and Akinori Yonezawa. A Simple Extension of Java Language for Controllable Transparent Migration and its Portable Implementation. In *Proceedings of Coordination Languages and Models (COORDINATION '99)*, Vol. 1594 of *Lecture Notes in Computer Science*, pp. 211–226, Amsterdam, The Netherlands, April 1999. Springer.
- Kenjiro Taura, Kunio Tabata, and Akinori Yonezawa. StackThreads/MP: Integrating Futures into Calling Standard. In *Proceedings of the seventh ACM SIGPLAN symposium on Principles and Practice of Parallel Programming (PPoPP '99)*, pp. 60–71, Atlanta, USA, May 1999.
- Hidehiko Masuhara and Akinori Yonezawa. Generating Optimized Residual Code in Run-Time Specialization. In *Technical Report on Partial Evaluation and Program Transformation Day (PE Day '99)*, p. 20 pages, Waseda, Tokyo, Japan, November 1999.
- Yoshihiro Oyama, Kenjiro Taura, and Akinori Yonezawa. Executing Parallel Programs with Synchronization Bottlenecks Efficiently. In *Proceedings of International Workshop on Parallel and Distributed Computing for Symbolic and Irregular Applications (PDSIA '99), July 5-7, 1999*, pp. 182–204, Sendai, Japan, April 2000. World Scientific.
- Hidehiko Masuhara and Akinori Yonezawa. An Object-Oriented Concurrent Reflective Language ABCL/R3: Its Meta-level Design and Efficient Implementation Techniques. In *Object-Oriented Parallel and Distributed Programming*, pp. 151–165, Paris, 2000. HERMES Science Publications.
- Hiroshi Yamauchi, Hidehiko Masuhara, Daisuke Hoshina, Tatsuro Sekiguchi, and Akinori Yonezawa. Wrapping Class Libraries for Migration-Transparent Resource

- Access by Using Compile-Time Reflection. In *Proceedings of Workshop on Reflective Middleware (RM2000)*, pp. 19–20, New York, April 2000.
- Yoshizumi Tanaka, Kenjiro Taura, Mitsuhsa Sato, and Akinori Yonezawa. Performance Evaluation of OpenMP Applications with Nested Parallelism. In *Proceedings of 5th International Workshop on Languages, Compilers and Runtime Systems for Scalable Computers (LCR 2000)*, Vol. 1915 of *Lecture Notes in Computer Science*, pp. 100–112, Rochester, NY, USA, May 2000. Springer.
 - Yoshihiro Oyama, Kenjiro Taura, and Akinori Yonezawa. Online Computation of Critical Paths for Multithreaded Languages. In *Proceedings of 5th International Workshop on High-Level Parallel Programming Models and Supportive Environments (HIPS 2000)*, Vol. 1800 of *Lecture Notes in Computer Science*, pp. 301–313, Cancun, Mexico, May 2000. Springer.
 - Masatomo Hashimoto and Akinori Yonezawa. MobileML: A Programming Language for Mobile Computation. In *Proceedings of 4th International Conference on Coordination Languages and Models (COORDINATION 2000)*, Vol. 1906 of *Lecture Notes in Computer Science*, pp. 198–215, Limassol, Cyprus, September 2000. Springer.
 - Takahiro Sakamoto, Tatsuou Sekiguchi, and Akinori Yonezawa. Bytecode Transformation for Portable Thread Migration in Java. In *Proceedings of Agent Systems, Mobile Agents, and Applications (ASA/MA 2000)*, Vol. 1882 of *Lecture Notes in Computer Science*, pp. 16–28, Zürich, Switzerland, September 2000. Springer.
 - Masatomo Hashimoto and Akinori Yonezawa. A Context-based Higher-Order Typed Language for Mobile Computation. In *Proceedings of the International Workshop on Mobile Objects/Code and Security (MOCS 2000)*, Tokyo, October 2000.
 - Hidehiko Masuhara, Yuuya Sugita, and Akinori Yonezawa. Dynamic Compilation of a Reflective Language Using Run-Time Specialization. In *Proceedings of International Symposium on Principles of Software Evolution (ISPSE 2000)*, pp. 125–134, Kanazawa, Japan, November 2000.
 - Tatsuou Sekiguchi, Takahiro Sakamoto, and Akinori Yonezawa. Portable Implementation of Continuation Operators in Imperative Languages by Exception Handling. In *Advances in Exception Handling Techniques (the book grow out of a ECOOP 2000 workshop)*, Vol. 2022 of *Lecture Notes in Computer Science*, pp. 217–233. Springer, 2001.
 - Toshio Endo, Kenjiro Taura, and Akinori Yonezawa. Predicting Scalability of Parallel Garbage Collectors on Shared Memory Multiprocessors. In *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS 2001)*, San Francisco, CA, USA, April 2001.

- Hidehiko Masuhara and Akinori Yonezawa. Run-time Bytecode Specialization: A Portable Approach to Generating Optimized Specialized Code. In *Proceedings of Second Symposium on Programs as Data Objects (PADO II)*, Vol. 2053 of *Lecture Notes in Computer Science*, pp. 138–154, Aarhus, Denmark, May 2001. Springer.
- Yoshihiro Oyama, Kenjiro Taura, and Akinori Yonezawa. Fusion of Concurrent Invocations of Exclusive Methods. In *Proceedings of 6th International Conference on Parallel Computing Technologies (PaCT 2001)*, Vol. 2127 of *Lecture Notes in Computer Science*, pp. 293–307, Novosibirsk, Russia, September 2001. Springer.
- Daisuke Hoshina, Eijiro Sumii, and Akinori Yonezawa. A Typed Process Calculus for Fine-Grained Resource Access Control in Distributed Computation. In *Proceedings of 4th International Symposium on Theoretical Aspects of Computer Software (TACS 2001)*, Vol. 2215 of *Lecture Notes in Computer Science*, pp. 64–81, Sendai, Japan, October 2001. Springer.
- Viet Ha Nguyen, Kenjiro Taura, and Akinori Yonezawa. Parallelizing Programs Using Access Traces. In *Proceedings of the 6th Workshop on Languages, Compilers, and Run-time Systems for Scalable Computers (LCR 2002)*, Washington D.C., USA, March 2002.
- Kenji Kaneda, Kenjiro Taura, and Akinori Yonezawa. Virtual Private Grid: A Command Shell for Utilizing Hundreds of Machines Efficiently. In *Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID 2002)*, pp. 212–219, Berlin, Germany, May 2002.
- Reynald Affeldt, Hidehiko Masuhara, Eijiro Sumii, and Akinori Yonezawa. Supporting Objects in Run-time Bytecode Specialization. In *Proceedings of ACM SIGPLAN ASIAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation (ASIA-PEPM 2002)*, pp. 50–60, Aizu, Japan, September 2002.
- Eric Y. Chen, Hitoshi Fuji, and Akinori Yonezawa. Solution Deployment on Multi-Provider Networks. In *OPENSIG 2002 Conference Proceedings*, Lexington, USA, October 2002.
- Yutaka Oiwa, Taturou Sekiguchi, Eijiro Sumii, and Akinori Yonezawa. Fail-Safe ANSI-C Compiler: An Approach to Making C Programs Secure (Progress Report). In *Proceedings of the International Symposium on Software Security (ISSS 2002), November, 2002*, Vol. 2609 of *Lecture Notes in Computer Science*, pp. 133–153, Tokyo, Japan, February 2003. Springer.
- Naoshi Tabuchi, Eijiro Sumii, and Akinori Yonezawa. Regular Expression Types for Strings in a Text Processing Language. In *Proceedings of Workshop on Types in Programming (TIP '02)*, Vol. 75, p. 19 pages, Dagstuhl, Germany, February 2003. Elsevier Science.

- Frédéric Peschanski, Jean-Pierre Briot, and Akinori Yonezawa. Fine-grained Dynamic Adaptation of Distributed Components. In *Proceedings of ACM/IFIP/USENIX International Middleware Conference (Middleware 2003)*, Vol. 2672 of *Lecture Notes in Computer Science*, pp. 123–142, Rio de Janeiro, Brazil, June 2003. Springer.
- Kenjiro Taura, Kenji Kaneda, Toshio Endo, and Akinori Yonezawa. Phoenix: a Parallel Programming Model for Accommodating Dynamically Joining/Leaving Resources. In *Proceedings of the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP 2003)*, pp. 216–229, San Diego, June 2003.
- Eric Y. Chen and Akinori Yonezawa. Federation of Network Service Providers and Its Applications. In *Proceedings of the Eighth IEEE Symposium on Computers and Communications (ISCC'2003)*, Kemer-Antalya, Turkey, July 2003.
- Toshiyuki Maeda and Akinori Yonezawa. Kernel Mode Linux: Toward an Operating System Protected by a Type Theory. In *Proceedings of 8th Asian Computing Science Conference*, Vol. 2896 of *Lecture Notes in Computer Science*, pp. 3–17, Mumbai, India, December 2003. Springer.
- Takashi Masuyama, Frédéric Peschanski, Yoshihiro Oyama, and Akinori Yonezawa. Mobile Scope: A Programming Language with Objective Mobility. In *Proceedings of the 2nd International Workshop on Mobile Distributed Computing (MDC '04)*, pp. 542–547, Tokyo, Japan, March 2004.
- Kenji Kaneda, Kenjiro Taura, and Akinori Yonezawa. Routing and Resource Discovery in Phoenix Grid-Enabled Message Passing Library. In *Proceedings of the 4th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2004)*, Chicago, USA, April 2004.
- Toshio Endo, Kenji Kaneda, Kenjiro Taura, and Akinori Yonezawa. High Performance LU Factorization for Non-dedicated Clusters. In *Proceedings of the 4th IEEE/ACM International Symposium on Cluster Computing and the Grid (CC-Grid 2004)*, Chicago, USA, April 2004.
- Akihito Nagata, Naoki Kobayashi, and Akinori Yonezawa. Region-Based Memory Management for a Dynamically-Typed Language. In *Proceedings of 2nd Asian Symposium on Programming Language and Systems (APLAS 2004)*, Vol. 3302 of *Lecture Notes in Computer Science*, pp. 229–245, Taipei, Taiwan, November 2004. Springer.
- Hideaki Tatsuzawa, Hidehiko Masuhara, and Akinori Yonezawa. Aspectual Caml: an Aspect-Oriented Functional Language. In *Proceedings of Foundations of Aspect-Oriented Languages (FOAL)*, Chicago, USA, March 2005.

- Yoshihiro Oyama, Koichi Onoue, and Akinori Yonezawa. Speculative Security Checks in Sandboxing Systems. In *Proceedings of The 1st International Workshop on Security in Systems and Networks (SSN2005)*, Denver, USA, April 2005.
- Toshiyuki Maeda and Akinori Yonezawa. Writing practical memory management code with a strictly typed assembly language. In *Proceedings of the 3rd Workshop on Semantics, Program Analysis, and Computing Environments for Memory Management (SPACE 2006)*, Charleston, South Carolina, USA, January 2006.
- Nicolas Marti, Reynald Affeldt, and Akinori Yonezawa. Verification of the heap manager of an operating system using separation logic. In *Proceedings of the 3rd Workshop on Semantics, Program Analysis, and Computing Environments for Memory Management (SPACE 2006)*, Charleston, South Carolina, USA, January 2006.
- Hiroshi Unno, Naoki Kobayashi, and Akinori Yonezawa. Combining Type-Based Analysis and Model Checking for Finding Counterexamples against Non-Interference. In *Proceedings of the ACM SIGPLAN Workshop on Programming Languages and Analysis for Security (PLAS 2006)*, Ottawa, Canada, June 2006.
- Khaled Ragab, Yoshihiro Oyama, and Akinori Yonezawa. K-Interleaving Rendezvous Overlay Network Construction Scheme. In *Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse (ICIS-COMSAR '06)*, Honolulu, Hawaii, USA, July 2006.
- Nicolas Marti, Reynald Affeldt, and Akinori Yonezawa. Formal Verification of the Heap Manager of an Operating System using Separation Logic. In *Proceedings of the 8th International Conference on Formal Engineering Methods (ICFEM 2006)*, Macao, China, October 2006.
- Koichi Onoue, Yoshihiro Oyama, and Akinori Yonezawa. A Virtual Machine Migration System Based on a CPU Emulator. In *Proceedings of the 1st International Workshop on Virtualization Technology in Distributed Computing (VTDC 2006)*, Tampa, Florida, USA, November 2006.
- Hidehiko Masuhara, Yusuke Endoh, and Akinori Yonezawa. A Fine-Grained Join Point Model for More Reusable Aspects. In *Proceedings of the 4th ASIAN Symposium on Programming Languages and Systems (APLAS 2006)*, Sydney, Australia, November 2006.
- Takahiro Kosakai, Toshiyuki Maeda, and Akinori Yonezawa. Compiling C Programs into a Strongly Typed Assembly Language. In *Proceedings of the 12th Asian Computing Science Conference (ASIAN 2007)*, Vol. 4846 of *Lecture Notes in Computer Science*, pp. 17–32, December 2007.

- Koichi Onoue, Yoshihiro Oyama, and Akinori Yonezawa. Control of system calls from outside of virtual machines. In *Proceedings of the 2008 ACM symposium on Applied computing (SAC 2008)*, pp. 2116–2221, Fortaleza, Brazil, March 2008.
- Motohiko Matsuda, Toshiyuki Maeda, and Akinori Yonezawa. Towards Design and Implementation of Model Checker for System Software. In *Proceedings of the 1st International Workshop on Software Technologies for Future Dependable Distributed Systems (STFSSD 2009)*, Tokyo, Japan, January 2009.
- Nan Dun, Kenjiro Taura, and Akinori Yonezawa. GMount: An Ad Hoc and Locality-Aware Distributed File System by Using SSH and FUSE. In *Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2009)*, pp. 188–195, Shanghai, China, May 2009.
- Toshiyuki Maeda and Akinori Yonezawa. Writing an OS Kernel in a Strictly and Statically Typed Language. In *Formal to Practical Security*, Vol. 5458 of *Lecture Notes in Computer Science*, pp. 181–197. Springer, 2009.
- Masaaki Shimizu and Akinori Yonezawa. Remote Process Execution and Remote I/O for Heterogeneous Processors in Cluster Systems. In *Proceedings of the 10th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2010)*, pp. 145–154, Melbourne, Australia, May 2010.
- Nan Dun, Kenjiro Taura, and Akinori Yonezawa. ParaTrac: A Fine-Grained Profiler for Data-Intensive Workflows. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing (HPDC2010)*, pp. 37–48, Chicago, USA, June 2010.
- Toshiyuki Maeda and Akinori Yonezawa. Typed Assembly Language for Implementing OS Kernels in SMP/Multi-Core Environments with Interrupts. In *Proceedings of the 5th International Workshop on Systems Software Verification (SSV 2010)*, Vancouver, Canada, October 2010.
- Nan Dun, Kenjiro Taura, and Akinori Yonezawa. Easy and Instantaneous Processing for Data-Intensive Workflows. In *Proceedings of the 3rd IEEE Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS2010)*, New Orleans, USA, November 2010.
- Junya Sawazaki, Toshiyuki Maeda, and Akinori Yonezawa. Implementing a Hybrid Virtual Machine Monitor for Flexible and Efficient Security Mechanisms. In *Proceedings of the 16th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2010)*, pp. 37–46, Tokyo, Japan, December 2010.
- Toshiyuki Maeda, Haruki Sato, and Akinori Yonezawa. Extended Alias Type System using Separating Implication. In *Proceedings of the 7th ACM SIGPLAN*

Workshop on Types in Language Design and Implementation (TLDI 2011), pp. 29–42, Austin, USA, January 2011.

- Tomohiro Suzuki, Kevin Pinte, Tom Van Cutsem, Wolfgang De Meuter, and Akinori Yonezawa. Programming Language Support for Routing in Pervasive Networks. In *Proceedings of the 8th International IEEE Workshop on Middleware and System Support for Pervasive Computing (PerWare 2011)*, Austin, USA, March 2011. To appear.

3.6 和文論文誌論文

- 米澤明憲. 述語論理における証明手続について – Theorem Prover TP-I –. 情報処理, Vol. 14, No. 2, pp. 106–113, February 1973.
- 米澤明憲. オブジェクト指向型プログラミングについて. コンピュータソフトウェア, Vol. 1, No. 1, pp. 29–41, April 1984.
- 大澤一郎, 米澤明憲. オブジェクト指向方式による対話理解システム. コンピュータソフトウェア, Vol. 2, No. 1, pp. 11–28, January 1985.
- 米澤明憲, 柴山悦哉, Jean-Pierre Briot, 本田康晃, 高田敏弘. オブジェクト指向に基づく並列情報処理モデル ABCM/1 とその記述言語 ABCL/1. コンピュータソフトウェア, Vol. 3, No. 3, pp. 9–23, April 1986.
- 渡部卓雄, 大澤一郎, 米澤明憲. 対話領域の独立性を指向した日本語対話理解システム. 人工知能学会論文誌, Vol. 2, No. 2, pp. 206–213, 1987.
- 外山滋, 米澤明憲. Event-Sequence Method for Qualitative Circuit Analysis. 人工知能学会誌, Vol. 3, No. 1, pp. 69–77, 1988.
- 米澤明憲. 並列オブジェクト指向言語 ABCL/1 による並列処理記述とその枠組みの研究 (招待論文). 電子情報通信学会論文誌, Vol. J71-D, No. 8, pp. 1415–1422, August 1988.
- 高田敏弘, 米澤明憲. 並列オブジェクト指向言語の分散環境における実現. コンピュータソフトウェア, Vol. 6, No. 1, pp. 17–29, January 1989.
- 村田剛志, 米澤明憲. 対話参加者の心的記述とその動的変化に関する枠組み. 人工知能学会誌, Vol. 8, No. 6, pp. 778–785, 1993.
- 一杉裕志, 松岡聡, 米澤明憲. 実行時カーネルのないリフレクティブな並列オブジェクト指向言語の実現方法. コンピュータソフトウェア, Vol. 11, No. 3, pp. 65–77, May 1994.
- 増原英彦, 松岡聡, 米澤明憲. 自己反映並列オブジェクト指向言語 ABCL/R2 の設計と実現. コンピュータソフトウェア, Vol. 11, No. 3, pp. 15–32, May 1994.
- 高橋伸, 松岡聡, 宮下健, 米澤明憲. アルゴリズムアニメーション作成システムにおける宣言的記述方法について. コンピュータソフトウェア, Vol. 11, No. 6, pp. 83–94, November 1994.
- 宮下健, 松岡聡, 高橋伸, 米澤明憲. 複数の視覚的例による直接操作インターフェイスの対話的实现. コンピュータソフトウェア, Vol. 11, No. 6, pp. 41–51, November 1994.
- 瀧武志, 米澤明憲. 伝達の含意を導く要素の分析とその利用法. 人工知能学会誌, Vol. 9, No. 6, pp. 899–907, November 1994.

- 増原英彦, 松岡聡, 米澤明憲. 並列自己反映言語システムの部分計算によるコンパイル技法. 情報処理学会論文誌, Vol. 37, No. 7, pp. 1290–1298, July 1996.
- 八杉昌宏, 松岡聡, 米澤明憲. スレッドベース実行における積極的データ転送のための Plan-Do 型コンパイル技法とその評価. 情報処理学会論文誌, Vol. 38, No. 9, pp. 1840–1848, September 1997.
- 八杉昌宏, 松岡聡, 米澤明憲. ABCL/EM-4: データ駆動並列計算機上の並列オブジェクト指向言語処理系の実装と評価. 情報処理学会論文誌, Vol. 38, No. 9, pp. 1790–1799, September 1997.
- 高橋俊行, 石川裕, 佐藤三久, 米澤明憲. メタレベル機能によるクラスライブラリ最適化手法. 情報処理学会論文誌, Vol. 39, No. 6, pp. 1681–1690, June 1998.
- 増原英彦, 米澤明憲. Reasoning を容易にする並列自己反映言語のメタオブジェクトの設計. コンピュータソフトウェア, Vol. 15, No. 4, pp. 62–66, July 1998.
- 今井健男, 山本泰宇, 遠藤敏夫, 田浦健次郎, 米澤明憲. DisC: ごみ集め機構を備えた C++ の分散記憶並列拡張言語. 情報処理学会論文誌 プログラミング, Vol. 40, No. SIG01, pp. 42–56, February 1999.
- 外山純生, 大山恵弘, 田浦健次郎, 米澤明憲. 明示的なタスク配置指定が可能な遅延タスク生成に基づく動的負荷分散方法. 情報処理学会論文誌, Vol. 40, No. SIG01 (PRO2), pp. 1–12, February 1999.
- 細部博史, 松岡聡, 米澤明憲. HiRise: GUI 構築のためのインクリメンタルな制約解消系. コンピュータソフトウェア, Vol. 16, No. 6, pp. 33–45, November 1999.
- 田中義純, 田浦健次郎, 米澤明憲. OpenMP におけるネストした並列性の実装と評価. 情報処理学会論文誌 プログラミング, Vol. 41, No. SIG02, pp. 54–64, March 2000.
- 多賀奈由太, 関口龍郎, 米澤明憲. 通常の実行効率をほとんど損わないスレッドマイグレーションが可能な C++. 情報処理学会論文誌 プログラミング, Vol. 41, No. SIG02, pp. 41–53, March 2000.
- 田浦健次郎, 米澤明憲. 分散記憶並列計算機における局所ごみ集めのスケジュール方式について. 情報処理学会論文誌, Vol. 41, No. 5, pp. 1490–1499, May 2000.
- 山本泰宇, 田浦健次郎, 米澤明憲. 分散メモリ並列計算機上での参照カウントと分散マーキングの実装および性能比較. 情報処理学会論文誌, Vol. 41, No. 5, pp. 1480–1489, May 2000.
- 田浦健次郎, 米澤明憲. 最小限のコンパイラサポートによる細粒度マルチスレッディング - 効率的なマルチスレッド言語を実装するためのコスト効率の良い方法. 情報処理学会論文誌, Vol. 41, No. 5, pp. 1459–1469, May 2000.

- 大山恵弘, 田浦健次朗, 米澤明憲. 同期ボトルネックが存在する並列プログラムの効率の実行. 情報処理学会論文誌, Vol. 41, No. 5, pp. 1448–1458, May 2000.
- 松原克弥, 板橋一正, 森山豊, 染谷祐一, 加藤和彦, 関口龍郎, 米澤明憲. 動的双方向変換技術に基づいた異機種オブジェクトモビリティの実現法. 情報処理学会論文誌, Vol. 41, No. 6, pp. 1651–1664, June 2000.
- 大山恵弘, 田浦健次朗, 米澤明憲. 排他的なメソッドの並行な呼び出しを融合する機構を持つ言語. 情報処理学会論文誌 プログラミング, Vol. 42, No. SIG02, pp. 13–25, February 2001.
- 関口龍郎, 米澤明憲. 異機種間モバイル計算のためのコード表現とその実装. 情報処理学会論文誌 プログラミング, Vol. 42, No. SIG02, pp. 1–12, February 2001.
- 上田陽平, 山本泰宇, 関口龍郎, 米澤明憲. アセンブリ言語レベルでの異種計算機間のヒープとスタックの共有機構. 情報処理学会論文誌 プログラミング, Vol. 42, No. SIG03, pp. 27–39, March 2001.
- 遠藤敏夫, 田浦健次朗, 米澤明憲. 共有メモリ並列計算機上の並列ガーベージコレクタの性能予測. コンピュータソフトウェア, Vol. 18, No. 2, pp. 54–58, March 2001.
- 橋本政朋, 米澤明憲. 移動計算のためのプログラミング言語 MobileML. コンピュータソフトウェア, Vol. 18, No. 2, pp. 2–19, March 2001.
- 板橋一正, 松原克弥, 森山豊, 染谷祐一, 加藤和彦, 関口龍郎, 米澤明憲. 仮想機械独立なアプレットシステムの実現. 電子情報通信学会論文誌, Vol. J84-D1, No. 6, pp. 639–649, June 2001.
- 関口龍郎, 大岩寛, 米澤明憲. オブジェクト指向言語によって記述された, 携帯電話, PDA のアプリケーションプログラム圧縮方式. コンピュータソフトウェア, Vol. 19, No. 1, pp. 1–9, January 2002.
- 速水雄太, 田浦健次朗, 米澤明憲. Java バイトコード変換による細粒度 CPU 資源管理. 情報処理学会論文誌 プログラミング, Vol. 43, No. SIG03, pp. 41–51, March 2002.
- 大岩寛, 住井英二郎, 米澤明憲. 安全性を保証する ANSI-C 実行系の実装手法. コンピュータソフトウェア, Vol. 19, No. 3, pp. 39–44, May 2002.
- 田淵直, 住井英二郎, 米澤明憲. テキスト処理言語における文字列のための正規表現型. 情報処理学会論文誌 プログラミング, Vol. 44, No. SIG02, pp. 1–12, February 2003.
- 住井英二郎, 大根田裕一, 米澤明憲. 例外処理機構を備えた命令型言語の CPS 変換とその定式化. 情報処理学会論文誌 : プログラミング, Vol. 45, No. 12, pp. 67–82, 2004.

- 住井英二郎, 立沢秀晃, 米澤明憲. セキュリティプロトコルの略式記法から spi 計算への変換. 情報処理学会論文誌 : プログラミング, Vol. 45, No. 12, pp. 1–10, 2004.
- 金田憲二, 大山恵弘, 米澤明憲. 単一システムイメージを提供するための仮想マシンモニタ. 情報処理学会論文誌 コンピューティングシステム, Vol. 47, No. 3, pp. 27–39, March 2006.
- 飯塚大輔, 前田俊行, 米澤明憲. CPU の条件実行機能に対応した型付きアセンブリ言語. 情報処理学会論文誌 プログラミング, Vol. 1, No. 2, pp. 9–27, 2008.
- 西川賀樹, 大山恵弘, 米澤明憲. プロセスレベルの仮想化を用いた大規模分散システムテストベッド. 情報処理学会論文誌 コンピューティングシステム, Vol. 1, No. 2, pp. 144–156, 2008.
- 清水正明, 小笠原克久, 船生真紀子, 米澤明憲. ヘテロジニアスシステム向けリモートプロセス管理機能. 情報処理学会論文誌 コンピューティングシステム, Vol. 49, No. SIG2, pp. 10–19, 2008.
- 尾上浩一, 大山恵弘, 米澤明憲. システムコール制御に基づく仮想マシン間サンドボックスシステム. 情報処理学会論文誌 コンピューティングシステム, Vol. 2, No. 1, pp. 33–52, 2009.
- 尾上浩一, 大山恵弘, 米澤明憲. アプリケーションデータを保護するための VMM に基づくアーキテクチャ. 情報処理学会論文誌 コンピューティングシステム, Vol. 2, No. 3, pp. 173–188, 2009.

3.7 解説

- 米澤明憲. ACTOR 理論について. 情報処理, Vol. 20, No. 7, pp. 580–589, July 1979.
- 米澤明憲. データ・フロー計算機 / 関数型プログラミングへのコメント. 情報処理, Vol. 21, No. 7, p. 797, July 1980.
- 佐渡一広, 米澤明憲. 抽象データ型言語. 情報処理, Vol. 22, No. 6, pp. 525–530, June 1981.
- 米澤明憲. 関数型計算モデル. 情報処理, Vol. 24, No. 2, pp. 113–122, February 1983.
- 柴山悦哉, 松田裕幸, 米澤明憲. 並列オブジェクト指向言語 ABCL による在庫管理システムの記述. 情報処理, Vol. 26, No. 5, pp. 460–468, May 1985.
- 米澤明憲. オブジェクト指向計算の現状と展望. 情報処理, Vol. 29, No. 4, pp. 290–294, April 1988.
- Akinori Yonezawa. The Real World Computing Program: MITI's Next Computer Research Initiative. *Science*, Vol. 258, pp. 581–582, October 1992.
- 中谷明弘, 山本健二, 米澤明憲. 超並列計算機を用いた RNA 2 次構造の予測と視覚化. 情報処理, Vol. 37, No. 10, pp. 946–951, October 1996.
- 米澤明憲. 安全な情報社会基盤を構築するために. 岩波書店「科学」, Vol. 74, No. 2, pp. 174–177, February 2004.
- 米澤明憲. 私のソフトウェア研究. コンピュータソフトウェア, Vol. 21, No. 5, pp. 20–31, August 2004.
- 米澤明憲, 前田俊行. 並列オブジェクトによる大規模システムの実現 –Second Life システム, Twitter システム, 分子動力学アプリの場合–. 電子情報通信学会論文誌 D, Vol. J93-D, No. 6, pp. 656–662, June 2010.

3.8 国内会議論文

- 米澤明憲. Theorem Prover TP-I. 情報処理学会第 12 回大会予稿集, pp. 239–240, 1971.
- 米澤明憲. 並列プログラムにおける抽象データ型の形式的仕様について. 昭和 53 年度情報処理学会第 19 回全国大会予稿集, 1978.
- 米澤明憲. ACTOR 理論と分散型処理系. 情報処理学会 研究報告 ソフトウェア工学, July 1978.
- 米澤明憲. 高度な並列性を内部にもつソフトウェアシステムに対する仕様及び検証技法について. 情報処理学会第 20 回プログラミングシンポジウム予稿集, 1979.
- 大澤一郎, 米澤明憲. オブジェクト指向方式による対話理解システム. 情報処理学会 研究報告 自然言語処理, No. 44, July 1984.
- 高田敏弘, 柴山悦哉, 米澤明憲. Common Lisp による並列オブジェクト指向言語の処理系の設計と実現. 日本ソフトウェア科学会第 3 回大会論文集, pp. 157–160, 東京工業大学, November 1986.
- 渡部卓雄, 大澤一郎, 米澤明憲. 対話領域の独立性を指向した日本語対話理解システム. 日本ソフトウェア科学会第 3 回大会論文集, pp. 1–4, 東京工業大学, November 1986.
- 大澤一郎, 米澤明憲. 並列オブジェクト指向言語による並列構文解析 (並行分散システムの記述と検証). 情報処理学会 研究報告 プログラミング, No. 19, December 1986.
- 外山滋, 米澤明憲. 動作系列別解析法による定性的回路解析とその実現. 情報処理学会 研究報告 知能と複雑系, pp. 41–48, March 1987.
- 渡部卓雄, 米澤明憲. Towards Reflection in an Object-Oriented Concurrent Language. 日本ソフトウェア科学会第 4 回大会論文集, pp. 339–342, 京都大学, November 1987.
- 高田敏弘, 米澤明憲. 並列オブジェクト指向言語の分散環境における実現. 第 4 回オブジェクト指向計算ワークショップ (WOOC '88), 1988.
- 渡部卓雄, 米澤明憲. アクターモデルにもとづいた並列自己反映計算モデル. 情報処理学会研究報告 (プログラミング言語研究会), 89-PL-23, No. 23, 1989.
- 渡部卓雄, 米澤明憲. 分散システムのための並列自己反映計算モデルにむけて: アクターモデルによるアプローチ. 日本ソフトウェア科学会第 6 回大会論文集, pp. 253–256, 九州大学, October 1989.
- 渡部卓雄, 米澤明憲. アクターモデルにもとづいた並列自己反映計算モデル. 電子情報通信学会技術研究報告, No. 330, pp. 77–85, December 1989.

- 松岡聡, 脇田建, 米澤明憲. 並列オブジェクト指向言語における継承について. 日本ソフトウェア科学会第7回大会論文集, pp. 65–68, 東京大学, October 1990.
- 松岡聡, 米澤明憲. 並列オブジェクト指向言語における Synchronization Constraints と継承について. 情報処理学会第41回全国大会論文集, 第5巻, pp. 28–29, 1990.
- 浅井健一, 松岡聡, 米澤明憲. マルチリスブの操作的意味論とその実装. 情報処理学会第41回全国大会論文集, 第5巻, pp. 8–9, 1990.
- 杉本徹, 米澤明憲. 対話理解のための心的状態の多重世界表現形式. 情報処理学会研究報告「自然言語処理」, No. 83, pp. 31–38, May 1991.
- 渡部卓雄, 松岡聡, 米澤明憲. 並行オブジェクト指向計算における自己反映計算の方式: Hybrid Group Reflective Architecture. 並列処理シンポジウム JSPP '91 論文集, pp. 421–428, May 1991.
- 古荘進一, 松岡聡, 米澤明憲. Concurrent Conservative Garbage Collection. 情報処理学会 研究報告 プログラミング, No. 3, July 1991.
- 高橋伸, 松岡聡, 米澤明憲, 鎌田富久. TRIP2: 抽象的データのグラフィカルな操作のためのプロトタイプシステム. 日本ソフトウェア科学会第8回大会論文集, pp. 577–580, 北海道大学, September 1991.
- 一杉裕志, 松岡聡, 渡部卓雄, 米澤明憲. 分散環境のための並列オブジェクト指向言語のリフレクティブアーキテクチャ. 日本ソフトウェア科学会第8回大会論文集, pp. 541–544, 北海道大学, September 1991.
- 八杉昌宏, 米澤明憲. N体問題の並列オブジェクト指向アルゴリズム. 日本ソフトウェア科学会第8回大会論文集, pp. 405–408, 北海道大学, September 1991.
- 古荘進一, 松岡聡, 米澤明憲. 共有メモリ型並列計算機上の Conservative Garbage Collection. 日本ソフトウェア科学会第8回大会論文集, pp. 89–92, 北海道大学, September 1991.
- 高橋伸, 松岡聡, 米澤明憲, 鎌田富久. 抽象的データと絵データ間の双方向変換に関する一般的枠組. 第7回ヒューマン・インタフェース・シンポジウム論文集 和文編, 第7巻, pp. 291–300, October 1991.
- 脇田建, 松岡聡, 米澤明憲. 並列オブジェクト指向言語のための並列制御機構. 情報処理学会 研究報告 データベース・システム プログラミング - 言語・基礎・実践 -, October 1991.
- 宮下健, 松岡聡, 高橋伸, 米澤明憲. 抽象的データに対する直接操作インターフェイスの例による実現. 情報処理学会 研究報告 プログラミング - 言語・基礎・実践 -, No. 10, 1992.
- 村田剛志, 米澤明憲. 対話参加者の心的記述とその動的変化に関する枠組. 情報処理学会 研究報告「自然言語処理」, No. 92-NL-87, pp. 67–74, January 1992.

- 浅井健一, 松岡聡, 米澤明憲. 制御部有限の CSP プログラムのモデル検査. 情報処理学会 研究報告 プログラミング, March 1992.
- 八杉昌宏, 松岡聡, 米澤明憲. ABCL/onEM-4: データ駆動計算機上の並列オブジェクト指向計算システムの高性能実装. 並列処理シンポジウム (JSPP '92) 論文集, pp. 171–178, 横浜, June 1992.
- 長塚雅明, 小林直樹, 松岡聡, 米澤明憲. 並列制約論理型言語の MIMD 型並列計算機上への実装. 並列処理シンポジウム (JSPP '92) 論文集, pp. 131–138, 横浜, June 1992.
- 田浦健次郎, 松岡聡, 米澤明憲. マルチコンピュータ上の並列オブジェクト指向言語の高効率な実装. 並列処理シンポジウム (JSPP '92) 論文集, 横浜, June 1992.
- 鳥澤健太郎, 米澤明憲. 対話領域に関する知識を獲得する対話システム – その設計と実現 –. 情報処理学会 研究報告 自然言語処理, No. 90, pp. 17–24, July 1992.
- Naoki Kobayashi and Akinori Yonezawa. Asynchronous Communication Model Based on Linear Logic. 日本ソフトウェア科学会第 9 回大会論文集, pp. 393–396, 慶応義塾大学湘南藤沢校舎, September 1992.
- 宮下健, 松岡聡, 高橋伸, 米澤明憲, 鎌田富久. 抽象的データに対する直接操作インターフェイスの例による実現. 日本ソフトウェア科学会第 9 回大会論文集, pp. 213–216, 慶応義塾大学湘南藤沢校舎, September 1992.
- 小林直樹, 松岡聡, 米澤明憲. 並列制約論理型言語における制約近似を用いた決定性解析. 日本ソフトウェア科学会第 9 回大会論文集, pp. 37–40, 慶応義塾大学湘南藤沢校舎, September 1992.
- 増原英彦, 松岡聡, 米澤明憲. 自己反映的並列オブジェクト指向言語 ABCL/R2 コンパイラについて. 情報処理学会 研究報告 プログラミング, No. 9, October 1992.
- 宮下健, 松岡聡, 高橋伸, 米澤明憲. 抽象的データに対する直接操作インターフェイスの例による実現. 情報処理学会 研究報告 プログラミング, No. 10, January 1993.
- 萩野達也, 徳田英幸, 斎藤信男, 松岡聡, 米澤明憲, 砂原秀樹, 多田好克, 柴山悦哉, 篠田陽一. 超並列計算機のためのオペレーティングシステムの構想. 情報処理学会 研究報告システムソフトウェアとオペレーティング・システム, No. 58, pp. 25–32, March 1993.
- Kenjiro Taura, Satoshi Matsuoka, and Akinori Yonezawa. Incorporating Locality Management into Garbage Collection in Massively Parallel Object-Oriented Languages. 並列処理シンポジウム (JSPP '93) 論文集, pp. 277–282, May 1993.
- 八杉昌宏, 松岡聡, 米澤明憲. ABCL/onEM-4: データ駆道計算機上の並列オブジェクト指向計算システムの高性能実装. オブジェクト指向コンピューティング I, 日本ソフトウェア科学会 WOOC '92, レクチャーノート/ソフトウェア学, 第 4 巻, pp. 289–304. 近代科学社, June 1993.

- 増原英彦, 松岡聡, 渡部卓雄, 米澤明憲. 自己反映計算の機能を持つ並列オブジェクト指向言語の高効率な実現方式. オブジェクト指向コンピューティング I, 日本ソフトウェア科学会 WOOC '92, レクチャーノート/ソフトウェア学, 第4巻, pp. 209–224. 近代科学社, June 1993.
- 一杉裕志, 松岡聡, 米澤明憲. 実行時カーネルのないリフレクティブな分散オブジェクト指向言語の設計と実現. オブジェクト指向コンピューティング I, 日本ソフトウェア科学会 WOOC '92, レクチャーノート/ソフトウェア学, 第4巻, pp. 31–47. 近代科学社, June 1993.
- 今野和浩, 長塚雅明, 小林直樹, 松岡聡, 米澤明憲. 並列制約論理型言語の最適化問題への応用. 日本ソフトウェア科学会第10回大会論文集, pp. 189–192, 早稲田大学, June-July 1993.
- 鎌田十三郎, 松岡聡, 米澤明憲. 並列計算機上の分散ガーベジコレクションのアルゴリズムとその性能評価. 日本ソフトウェア科学会第10回大会論文集, pp. 21–24, 早稲田大学, June-July 1993.
- Tatsurou Sekiguchi and Akinori Yonezawa. Inference of Message Arrival Order in Object-Oriented Concurrent Programs. 日本ソフトウェア科学会第10回大会論文集, pp. 17–20, 早稲田大学, June-July 1993.
- 細部博史, 宮下健, 高橋伸, 松岡聡, 米澤明憲, 鎌田富久. ユーザインターフェースで実時間のインタラクションを実現する制約解消法. 日本ソフトウェア科学会第10回大会論文集, pp. 77–80, 早稲田大学, June-July 1993.
- 高橋伸, 松岡聡, 宮下健, 米澤明憲, 鎌田富久. 宣言的記述によるアルゴリズムアニメーション作成のための枠組. 日本ソフトウェア科学会第10回大会論文集, pp. 105–108, 早稲田大学, June-July 1993.
- 鎌田十三郎, 松岡聡, 米澤明憲. 超並列計算機上の高効率な大域的ガーベジコレクション. 情報処理学会 研究報告 プログラミング, No. 13, August 1993.
- 宮下健, 松岡聡, 高橋伸, 米澤明憲. 複数の視覚的例による直接操作インターフェースの対話的实现. インタラクティブシステムとソフトウェア I-日本ソフトウェア科学会 WISS '93 (第1回インタラクティブシステムとソフトウェアに関するワークショップ論文集), 1993年12月, レクチャーノート/ソフトウェア学, 第7巻, pp. 241–248. 近代科学社, 1994.
- 高橋伸, 松岡聡, 宮下健, 米澤明憲. アルゴリズムアニメーション作成システムにおける宣言的記述方法について. インタラクティブシステムとソフトウェア I-日本ソフトウェア科学会 WISS '93 (第1回インタラクティブシステムとソフトウェアに関するワークショップ論文集), 1993年12月, レクチャーノート/ソフトウェア学, 第7巻, pp. 161–168. 近代科学社, 1994.

- 細部博史, 宮下健, 高橋伸, 松岡聡, 米澤明憲. 局所的な連立に基づく制約解消法. インタラクティブシステムとソフトウェア I-日本ソフトウェア科学会 WISS '93 (第1回インタラクティブシステムとソフトウェアに関するワークショップ論文集), 1993年12月, レクチャーノート/ソフトウェア学, 第7巻, pp. 49-56. 近代科学社, 1994.
- 八杉昌宏, 米澤明憲. データ駆動並列計算機上での並列オブジェクト指向言語による N 体問題アルゴリズムの性能評価に向けて. オブジェクト指向コンピューティング II, 日本ソフトウェア科学会 WOOC '93, March 18-20, 1993, レクチャーノート/ソフトウェア学, 第6巻, pp. 147-154. 近代科学社, April 1994.
- 増原英彦, 松岡聡, 米澤明憲. 自己反映並列オブジェクト指向言語によるメタレベルでのスケジューリング方式 - ホットスポットの検出と解消に向けて -. オブジェクト指向コンピューティング II, 日本ソフトウェア科学会 WOOC '93, March 18-20, 1993, レクチャーノート/ソフトウェア学, 第6巻, pp. 31-45. 近代科学社, April 1994.
- 長塚雅明, 今野和浩, 小林直樹, 松岡聡, 米澤明憲. 超並列計算機上の並列制約論理型言語 PARCS の柔軟な探索枝制御と高効率な枝刈り法. 並列処理シンポジウム (JSPP '94) 論文集, pp. 327-334, 工業技術院筑波研究センター, May 1994.
- 鎌田十三郎, 松岡聡, 米澤明憲. 超並列計算機上の高効率な大域的ガベージコレクション. 並列処理シンポジウム (JSPP '94) 論文集, pp. 33-40, 工業技術院筑波研究センター, May 1994.
- Kenjiro Taura, Satoshi Matsuoka, and Akinori Yonezawa. *StackThreads: An Abstract Machine for Scheduling Fine-Grain Threads on Stock CPUs.* 並列処理シンポジウム (JSPP '94) 論文集, pp. 25-32, 工業技術院筑波研究センター, May 1994.
- 増原英彦, 松岡聡, 米澤明憲. 高並列計算における動的資源管理のための自己反映並列オブジェクト指向言語. 1994年並列/分散/協調処理に関する『琉球』サマーマークワークショップ (SWoPP 琉球 '94), No. 94-PRG-18 in IPSJ SIG Notes, pp. 57-64, 沖縄, July 1994.
- 八杉昌宏, 松岡聡, 米澤明憲. スレッドベース実行における積極的データ転送のための Plan-Do 型コンパイル技法. 1994年並列/分散/協調処理に関する『琉球』サマーマークワークショップ (SWoPP 琉球 '94), No. 94-PRG-18 in IPSJ SIG Notes, pp. 9-16, 沖縄, July 1994.
- Naoki Kobayashi, Toshihiro Shimizu, and Akinori Yonezawa. Analysis on Communication for Asynchronous Concurrent Programming Languages (Extended Abstract). 日本ソフトウェア科学会第11回大会論文集, pp. 489-492, 大阪大学, October-November 1994.

- 中谷明弘, 小林直樹, 米澤明憲. 線形論理型言語の枠組 HACL に基づく型つき並列オブジェクト指向言語. 日本ソフトウェア科学会第 11 回大会論文集, pp. 485–488, 大阪大学, October–November 1994.
- 藤原頼晶, 田浦健次郎, 米澤明憲. 並列オブジェクト指向言語 ABCL/ f を用いた irregular なアプリケーションの並列化 - N 体問題の場合 -. 日本ソフトウェア科学会第 11 回大会論文集, pp. 145–148, 大阪大学, October–November 1994.
- 細部博史, 松岡聡, 米澤明憲. 不等式を含む制約階層の効率的解消法. インタラクティブシステムとソフトウェア III–日本ソフトウェア科学会 WISS'95 (第 3 回インタラクティブシステムとソフトウェアに関するワークショップ論文集), レクチャーノート/ソフトウェア学, 第 12 巻, pp. 123–132. 近代科学社, 1995.
- 大森直人, 松岡聡, 米澤明憲. 超並列計算用オブジェクト指向分散クラスライブラリの構築 - 階層的コレクションによる並列性と分散の分離 -. 並列処理シンポジウム (JSPP '95) 論文集, pp. 281–288, 福岡, May 1995.
- 増原英彦, 松岡聡, 米澤明憲. 並列自己反映言語システムの部分計算によるコンパイル技法. 並列処理シンポジウム (JSPP '95) 論文集, pp. 273–280, 福岡, May 1995.
- 細谷晴夫, 増原英彦, 田浦健次郎, 米澤明憲. プロセスとチャンネルに基づく並列言語のための部分計算. 1995 年並列/分散/協調処理に関する『別府』サマー・ワークショップ (SWoPP '95), No. 95-PRO-2 in IPSJ SIG Notes, pp. 73–80, 別府, August 1995.
- 増原英彦, 松岡聡, 米澤明憲. 並列オブジェクト指向言語 ABCL/ f のメタレベルアーキテクチャ. 1995 年並列/分散/協調処理に関する『別府』サマー・ワークショップ (SWoPP '95), No. 95-PRO-2 in IPSJ SIG Notes, pp. 65–72, 別府, August 1995.
- 佐藤直人, 松岡聡, 米澤明憲. 階層的コレクションに基づくオブジェクト指向分散ライブラリについて. 1995 年並列/分散/協調処理に関する『別府』サマー・ワークショップ (SWoPP '95), 情報処理学会 研究報告 ハイパフォーマンスコンピューティング, No. 95-HPC-57, pp. 37–48, 別府, August 1995.
- 中谷明弘, 田浦健次郎, 米澤明憲. 並列オブジェクト指向言語 ABCL/ f による RNA の 2 次構造予測. 1995 年並列/分散/協調処理に関する『別府』サマー・ワークショップ (SWoPP '95), 情報処理学会 研究報告 ハイパフォーマンスコンピューティング, No. 95-HPC-57, pp. 25–30, 別府, August 1995.
- 田浦健次郎, 米澤明憲. 並列オブジェクト指向言語 ABCL/ f による並列数値計算 - 有限要素法と多体問題による評価. 1995 年並列/分散/協調処理に関する『別府』サマー・ワークショップ (SWoPP '95), 情報処理学会 研究報告 ハイパフォーマンスコンピューティング, No. 57, pp. 19–24, 別府, August 1995.
- 今野和浩, 長塚雅明, 小林直樹, 松岡聡, 米澤明憲. 最適化問題への応用のための並列制約論理型言語の拡張. 情報処理学会第 51 回 (平成 7 年後期) 全国大会講演論文集 (5), pp. 77–78, 富山大学, September 1995.

- 浅井健一, 増原英彦, 米澤明憲. 副作用を含む関数型プログラムの部分評価に向けて. 情報処理学会 研究報告「プログラミング」, IPSJ SIG Notes, 95-PRO-4, No. 4, pp. 51–56, November 1995.
- 河内谷幸子, 米澤明憲. 複数の入出力デバイスを選択できる図形描画清書システム. 情報処理学会 研究報告 ヒューマンインタフェース, No. 63, November 1995.
- 細谷晴夫, 小林直樹, 米澤明憲. Partial Evaluation for Concurrent Languages and its Formalization. 第12回オブジェクト指向計算ワークショップ (WOOC '96) 論文集, 北陸先端科学技術大学院大学, March 1996.
- 五十嵐淳, 小林直樹, 米澤明憲. 並列言語におけるチャンネル送受信回数の静的型システムによる推論. 第12回オブジェクト指向計算ワークショップ (WOOC '96) 論文集, 北陸先端科学技術大学院大学, March 1996.
- 大山恵弘, 田浦健次郎, 米澤明憲. 動的なスレッド生成をサポートする言語のコンパイル技法. 1996年並列/分散/協調処理に関する『秋田』サマー・ワークショップ (SWoPP '96), 情報処理学会 研究報告 プログラミング, No. 96-PRO-8, pp. 115–120, 秋田, August 1996.
- 高橋俊行, 石川裕, 佐藤三久, 米澤明憲. メタレベル機能による並列プログラミング. 1996年並列/分散/協調処理に関する『秋田』サマー・ワークショップ (SWoPP '96), 情報処理学会 研究報告 プログラミング, No. 96-PRO-8, pp. 79–84, 秋田, August 1996.
- 遠藤敏夫, 田浦健次郎, 米澤明憲. Portableでrobustなglobal garbage collectorの構築について. 1996年並列/分散/協調処理に関する『秋田』サマー・ワークショップ (SWoPP '96), 情報処理学会 研究報告 プログラミング, No. 96-PRO-8, pp. 7–12, 秋田, August 1996.
- 大山恵弘, 田浦健次郎, 米澤明憲. 並行プロセス計算に基づく言語の効率的なコンパイル技法. 第13回オブジェクト指向計算ワークショップ (WOOC '97), 箱根, March 1997.
- 遠藤敏夫, 田浦健次郎, 米澤明憲. 共有メモリ並列計算機のための負荷分散を行うガーベージコレクタとその評価. 第13回オブジェクト指向計算ワークショップ (WOOC '97), 箱根, March 1997.
- 佐藤直人, 松岡聡, 米澤明憲. Compositional Traverser: a Framework for distributed object collections in parallel object-oriented programming. 第13回オブジェクト指向計算ワークショップ (WOOC '97), 箱根, March 1997.
- 田端邦男, 田浦健次郎, 米澤明憲. CプログラムにおけるLazy Task Creation. 1997年並列/分散/協調処理に関する『火の国』サマー・ワークショップ (SWoPP '97), 情報処理学会研究報告, No. 97-PRO-14, pp. 79–84, 阿蘇, August 1997.

- 山本泰宇, 田浦健次郎, 米澤明憲. 分散メモリ並列計算機における Reference Count GC と Mark and Sweep GC の比較. 1997 年並列/分散/協調処理に関する『火の国』サマー・ワークショップ (SWoPP '97), 情報処理学会研究報告, No. 97-PRO-14, pp. 109–114, 阿蘇, August 1997.
- 高橋俊行, 石川裕, 佐藤三久, 米澤明憲. メタレベル機能が支援するハイパフォーマンスオブジェクト指向計算. 1997 年並列/分散/協調処理に関する『火の国』サマー・ワークショップ (SWoPP '97), 情報処理学会研究報告, No. 97-PRO-14, pp. 37–42, 阿蘇, August 1997.
- 増原英彦, 米澤明憲. Reasoning を容易にする並列自己反映言語のメタオブジェクトの設計. 日本ソフトウェア科学会第 14 回大会論文集, 石川県ハイテク交流センター, September 1997.
- 細部博史, 松岡聡, 米澤明憲. 階層線形系を用いた効率的な制約階層解消法. インタラクティブシステムとソフトウェア V—日本ソフトウェア科学会 WISS'97 (第 5 回インタラクティブシステムとソフトウェアに関するワークショップ論文集), レクチャーノート/ソフトウェア学, 第 18 巻, pp. 129–134. 近代科学社, 1997.
- 細部博史, 松岡聡, 米澤明憲. HiRise: GUI 構築のためのインクリメンタルな制約解消系. インタラクティブシステムとソフトウェア VI—日本ソフトウェア科学会 WISS'98 (第 6 回インタラクティブシステムとソフトウェアに関するワークショップ論文集), レクチャーノート/ソフトウェア学, 第 21 巻, pp. 73–82. 近代科学社, 1998.
- 大山恵弘, 田浦健次郎, 遠藤敏夫, 米澤明憲. 細粒度スレッド生成をサポートする言語の共有メモリ並列計算機上での実装とその性能評価. 第 1 回プログラミングおよび応用のシステムに関するワークショップ (SPA '98) 論文集, 草津, March 1998.
- 山本泰宇, 田浦健次郎, 米澤明憲. 分散メモリ並列計算機上での参照カウントと大域 GC の実装及び性能比較. 並列処理シンポジウム (JSPP '98) 論文集, 名古屋, June 1998.
- 遠藤敏夫, 田浦健次郎, 米澤明憲. 大規模共有メモリ並列マシンにおけるスケラブルなマークスイープ法ガーベジコレクタ. 日本ソフトウェア科学会第 15 回大会論文集, pp. 341–344, 電気通信大学, September 1998.
- 関口龍郎, 米澤明憲. マイグレーションの制御としての側面. 日本ソフトウェア科学会第 15 回大会論文集, pp. 281–284, 電気通信大学, September 1998.
- 大岩寛, 田浦健次郎, 米澤明憲. Java 仮想マシンへの整数と参照の相互変換の導入. 第 2 回プログラミングおよび応用のシステムに関するワークショップ (SPA '99) 論文集, 熱川, March 1999.
- 増原英彦, 米澤明憲. Java バイトコード上での実行時プログラム特化. 第 2 回プログラミングおよび応用のシステムに関するワークショップ (SPA '99) 論文集, 熱川, March 1999.

- 後藤礼史, 田浦健次郎, 米澤明憲. 保護属性の動的変更が可能なソフトウェアによる細粒度メモリ保護. 並列処理シンポジウム (JSPP '99) 論文集, pp. 245–252, つくば, June 1999.
- 大山恵弘, 田浦健次郎, 米澤明憲. 潜在的なボトルネックが存在する並列プログラムの効率的実行. 並列処理シンポジウム (JSPP '99) 論文集, pp. 95–102, つくば, June 1999.
- 橋本政朋, 米澤明憲. 移動計算のためのプログラミング言語 MobileML. 第2回プログラミングおよびプログラミング言語ワークショップ (PPL 2000) 論文集, 浜名湖, March 2000.
- 関口龍郎, 米澤明憲. 異機種間モバイル計算のためのコード表現とその実装. 第3回プログラミングおよび応用のシステムに関するワークショップ (SPA2000) 論文集, 浜名湖, March 2000.
- 遠藤敏夫, 田浦健次郎, 米澤明憲. 共有メモリ並列計算機上の並列ガーベージコレクタの性能予測. 日本ソフトウェア科学会第17回大会論文集, 東京工科大学, September 2000.
- 星名大輔, 関口龍郎, 山内裕史, 米澤明憲. Mobile JDK: モバイルエージェント環境における計算機資源への移動透過な操作が可能なクラスライブラリ. 情報処理学会論文誌 プログラミング, Vol. 41, No. SIG 9(PRO 8), p. 111, November 2000. (発表概要).
- 板橋一正, 松原克弥, 森山豊, 染谷祐一, 加藤和彦, 関口龍郎, 米澤明憲. Native Applet: ネイティブコード実行に基づいた WWW 上のモバイルコード実行方式. 情報処理学会論文誌 プログラミング, Vol. 41, No. SIG 9(PRO 8), p. 109, November 2000. (発表概要).
- 浜中信行, 山本泰宇, 米澤明憲. 環状分散ゴミ集めのための局所ゴミ集めの最小限の拡張. 情報処理学会論文誌 プログラミング, Vol. 42, No. SIG 3(PRO 10), p. 77, March 2001. (発表概要).
- 後藤礼史, 田浦健次郎, 米澤明憲. Secure Shared Memory: オブジェクトを効率良く安全に共有するためのモデル. 第3回プログラミングおよびプログラミング言語ワークショップ (PPL 2001) 論文集, 京都, March 2001.
- 関口龍郎, 大岩寛, 米澤明憲. オブジェクト指向言語によって記述された、携帯電話、PDA のアプリケーションプログラム圧縮方式. 第3回プログラミングおよびプログラミング言語ワークショップ (PPL 2001) 論文集, 京都, March 2001.
- Yutaka Oiwa, Hidehiko Masuhara, and Akinori Yonezawa. DynJava: Type Safe Dynamic Code Generation in Java. 第3回プログラミングおよびプログラミング言語ワークショップ (PPL 2001) 論文集, 京都, March 2001.

- 田中義純, 田浦健次朗, 米澤明憲. 定期的なポーリングを保証するアルゴリズム. 並列処理シンポジウム (JSPP 2001) 論文集, 京都, June 2001.
- 遠藤敏夫, 田浦健次朗, 米澤明憲. 局所性を考慮した共有メモリ並列計算機上の並列 BIBOP 型メモリアロケータ. 並列処理シンポジウム (JSPP 2001) 論文集, 京都, June 2001.
- 金田憲二, 田浦健次朗, 米澤明憲. Virtual Private Grid (VPG): 遠隔計算機を効率的に利用するシェル. 2001年並列/分散/協調処理に関する『沖縄』サマー・ワークショップ (SWoPP 2001), 情報処理学会研究報告, No. 2001-HPC-87, pp. 183–188, 沖縄, July 2001.
- Reynald Affeldt, Hidehiko Masuhara, Eijiro Sumii, and Akinori Yonezawa. Supporting Objects in Run-time Bytecode Specialization. 日本ソフトウェア科学会第18回大会論文集, 公立はこだて未来大学, September 2001.
- 洪淳祥, 田浦健次朗, 米澤明憲. 動的に増減する資源を利用した広域分散 web crawling. 日本ソフトウェア科学会第18回大会論文集, 公立はこだて未来大学, September 2001.
- 大岩寛, 住井英二郎, 米澤明憲. 安全性を保証する ANSI-C 実行系の実装手法. 日本ソフトウェア科学会第18回大会論文集, 公立はこだて未来大学, September 2001.
- 浜中信行, 住井英二郎, 小林直樹, 米澤明憲. Java バイトコードにおけるオブジェクト使用解析のための型システム. 第4回プログラミングおよびプログラミング言語ワークショップ (PPL 2002) 論文集, 別府, March 2002.
- 前田俊行, 住井英二郎, 米澤明憲. Linux/TAL: 型付きアセンブリプログラムのカーネルモード実行方式. 第4回プログラミングおよびプログラミング言語ワークショップ (PPL 2002) 論文集, pp. 62–73, 別府, March 2002.
- エリックチェン, 柏大, 富士仁, 米澤明憲. Moving Firewall における DDoS 攻撃対策システムの評価. 電子情報通信学会技術研究報告, No. CS2002-76, pp. 73–77, September 2002.
- 増山隆, 住井英二郎, 米澤明憲. C++テンプレートを分割コンパイルするためのアプローチ. 情報処理学会論文誌 プログラミング, Vol. 44, No. SIG 2(PRO 16), p. 42, February 2003. (発表概要).
- 小林義徳, 遠藤敏夫, 田浦健次朗, 米澤明憲. マークスイープとコピーの混合による効率的なゴミ集め. 第5回プログラミングおよびプログラミング言語ワークショップ (PPL 2003) 論文集, 富士, March 2003.
- 速水雄太, 田浦健次朗, 米澤明憲. 消費 CPU 資源を細粒度に計測するアルゴリズム. 第5回プログラミングおよびプログラミング言語ワークショップ (PPL 2003) 論文集, 富士, March 2003.

- 末永幸平, 大岩寛, 住井英二郎, 米澤明憲. Fail-Safe C のためのインターフェイス定義言語. 第 5 回プログラミングおよびプログラミング言語 ワークショップ (PPL 2003) 論文集, 富士, March 2003.
- 永田章人, 小林直樹, 米澤明憲. 動的型付き言語のためのリージョン推論に基づくメモリ管理. 日本ソフトウェア科学会第 20 回記念大会論文集, pp. 31–35, 愛知県立大学, September 2003.
- 増山隆, フレデリックペシャンスキ, 大山恵弘, 米澤明憲. MobileScope: 透明な移動機能を備えた分散コンポーネント言語. 日本ソフトウェア科学会第 20 回記念大会論文集, pp. 196–200, 愛知県立大学, September 2003.
- 清木昌, 大山恵弘, 米澤明憲. モデル検査のゲームシナリオへの適用. 第一回ディペンドブルソフトウェアワークショップ (DSW04), pp. 39–48, 東京, February 2004.
- 吉野寿宏, 大山恵弘, 米澤明憲. 自己修復型リファレンスモニタの設計と実装. 第 7 回プログラミングおよび応用のシステムに関するワークショップ (SPA2004) 論文集, 長野, March 2004.
- 住井英二郎, 立沢秀晃, 米澤明憲. セキュリティプロトコルの略式表現から spi 計算への変換. 情報処理学会第 48 回プログラミング研究会, 東京工業大学, March 2004.
- 住井英二郎, 大根田裕一, 米澤明憲. 例外処理機構を備えた命令型言語の cps 変換とその定式化. 情報処理学会第 48 回プログラミング研究会, 東京工業大学, March 2004.
- 尾上浩一, 大山恵弘, 米澤明憲. サンドボックスシステムにおける投機的な安全性検査. 情報処理学会 研究報告 2004-OS-96, pp. 1–6, 沖縄, June 2004.
- 金田憲二, 田浦健次郎, 米澤明憲. 接続を動的に制御するメッセージパッシングシステム. 2004 年並列 / 分散 / 協調処理に関する『青森』サマー・ワークショップ (SWoPP 青森 2004), 青森, July-August 2004.
- Reynald Affeldt, Naoki Kobayashi, and Akinori Yonezawa. Verification of concurrent programs using the Coq theorem prover: a case study. 2004 年並列 / 分散 / 協調処理に関する『青森』サマー・ワークショップ (SWoPP 青森 2004), 青森, July-August 2004.
- 横山陽介, 大山恵弘, 米澤明憲. SoftwarePot へのチェックポイント機構の導入. 第 3 回情報科学技術フォーラム (FIT 2004) 予稿集, 同志社大学, September 2004.
- 大山恵弘, 加藤和彦, 米澤明憲. 仮想計算環境における最適化を支援するための API. 日本ソフトウェア科学会第 21 回大会論文集, 東京工業大学, September 2004.
- 立沢秀晃, 増原英彦, 米澤明憲. 関数型プログラミングのためのアスペクト指向言語. 日本ソフトウェア科学会第 21 回大会論文集, 東京工業大学, September 2004.

- 小林義徳, Eric Y. Chen, 大山恵弘, 米澤明憲. オクトパス攻撃から web サーバーを防御する手法. 第 2 回ディペンダブルソフトウェアワークショップ (DSW '05), pp. 87–95, 東京大学, January 2005.
- 大根田裕一, 増原英彦, 米澤明憲. 値間依存性に基づくポイントカット記述のためのバイトコード変換. 第 8 回プログラミングおよび応用のシステムに関するワークショップ (SPA 2005), 伊香保, March 2005.
- 尾上浩一, 大山恵弘, 米澤明憲. Quasar: Cpu エミュレータ qemu を利用した移動計算システム. 第 8 回プログラミングおよび応用のシステムに関するワークショップ (SPA 2005), 伊香保, March 2005.
- 金田憲二, 大山恵弘, 米澤明憲. 単一システムイメージを提供するための仮想マシンモニタ. 第 17 回コンピュータシステムシンポジウム (ComSys2005), pp. 3–12, 筑波大学, November 2005.
- 島本大輔, 大山恵弘, 米澤明憲. System service 監視による windows 向け異常検知システム. 先進的計算基盤システムシンポジウム (SAC SIS 2006), 大阪, May 2006.
- Nicolas Marti, Reynald Affeldt, and Akinori Yonezawa. Model-checking of a Multi-threaded Operating System. In *Proceedings of the 23rd Workshop of the Japan Society for Software Science and Technology*, Tokyo, Japan, September 2006.
- 前田俊行, 米澤明憲. 割込みに対応した型付きアセンブリ言語. 第 5 回 ディペンダブルシステムワークショップ, 函館, July 2007.
- 吉野寿宏, 前田俊行, 米澤明憲. 低級言語のプログラムを検証するための共通言語の設計とプログラム変換. 第 5 回 ディペンダブルシステムワークショップ, 函館, July 2007.
- 尾上浩一, 大山恵弘, 米澤明憲. 仮想マシンモニタによる仮想マシン内プロセスの制御. 2007 年並列/分散/協調処理に関する「旭川」サマー・ワークショップ, 旭川, August 2007.
- 西川賀樹, 大山恵弘, 米澤明憲. OS 資源ビューの仮想化を用いた分散システムテストベッド. 第 6 回情報科学技術フォーラム (FIT2007), 中京大学, September 2007.
- 清水正明, 小笠原克久, 船生真紀子, 米澤明憲. ヘテロジニアスシステム向けリモートプロセス管理機能. 第 19 回コンピュータシステムシンポジウム (ComSys2007), 東京, November 2007.
- 西川賀樹, 大山恵弘, 米澤明憲. プロセスレベルの仮想化を用いた大規模分散システムテストベッド. 第 6 回先進的計算基盤システムシンポジウム (SAC SIS 2008), pp. 195–204, 筑波, June 2008.
- 野尻隆弘, 前田俊行, 米澤明憲. モデル検査器を用いた依存型システムのための配列境界の推論. 第 6 回ディペンダブルシステムワークショップ (DSW 2008), 函館, July 2008.

- 前田俊行, 米澤明憲. SMP・マルチコアに対応した型付きアセンブリ言語. 第6回ディペンダブルシステムワークショップ (DSW 2008), 函館, July 2008.
- 松田元彦, 前田俊行, 米澤明憲. 大規模システムソフトウェアのモデル検査器の設計と実装. 2008年並列/分散/協調処理に関する『佐賀』サマー・ワークショップ, 佐賀, August 2008.
- 尾上浩一, 大山恵弘, 米澤明憲. アプリケーションデータを保護するためのVMMに基づくアーキテクチャ. 第7回先進的計算機版システムシンポジウム (SACSYS 2009), May 2009.
- 野尻隆弘, 前田俊行, 米澤明憲. モデル検査器SPINにおける安全性検証手法の並列化. 第7回ディペンダブルシステムワークショップ (DSW 2009), 函館, July 2009.
- 渡邊裕貴, 前田俊行, 米澤明憲. 配列のためのホーア型理論の拡張. 第7回ディペンダブルシステムワークショップ (DSW 2009), 函館, July 2009.
- 清水正明, 米澤明憲. ヘテロジニアスクラスタ向けシングルシステムI/O機能. 第21回コンピュータシステムシンポジウム (ComSys2009), 筑波大学, November 2009.
- 前田俊行, 米澤明憲. メモリコンシステンシモデルを考慮した型付きアセンブリ言語. 第8回ディペンダブルシステムワークショップ (DSW 2010), 函館, July 2010.

3.9 技術報告

- Akinori Yonezawa and Carl Hewitt. Symbolic Evaluation Using Conceptual Representations for Programs with Side-Effects. Technical Report 399, MIT Artificial Intelligence Laboratory, December 1976. AI-Memo.
- Akinori Yonezawa. Specification and Verification Techniques for Parallel Programs Based on Message Passing Semantics. Technical Report TR-191, MIT Laboratory for Computer Science, December 1977. Ph. D Thesis.
- Akinori Yonezawa, Hiroyuki Matsuda, and Etsuya Shibayama. An Object Oriented Approach for Concurrent Programming. Technical Report C-63, Department of Information Science, Tokyo Institute of Technology, 1984.
- Akinori Yonezawa, Hiroyuki Matsuda, and Etsuya Shibayama. Discrete Event Simulation Based on an Object-Oriented Parallel Computation Model. Technical Report C-64, Department of Information Science, Tokyo Institute of Technology, 1984.
- Satoshi Matsuoka, Ken Wakita, and Akinori Yonezawa. Synchronization Constraints With Inheritance: What Is Not Possible - So What Is? Technical Report is-90-010, The University of Tokyo, Faculty of Science, Department of Information Science, 1990.
- Yuuji Ichisugi and Akinori Yonezawa. Distributed Garbage Collection Using Group Reference Counting. Technical Report is-90-014, The University of Tokyo, Faculty of Science, Department of Information Science, 1990.
- Naoki Kobayashi and Akinori Yonezawa. Asynchronous Communication Model Based on Linear Logic. Technical Report TR92-05, The University of Tokyo, Faculty of Science, Department of Information Science, 1992.
- Masahiro Yasugi and Akinori Yonezawa. An Object-Oriented Parallel Algorithm for the Newtonian N-Body Problem. Technical Report TR92-06, The University of Tokyo, Faculty of Science, Department of Information Science, 1992.
- Naoki Kobayashi and Akinori Yonezawa. Logical, Testing, and Observation Equivalence for Processes in a Linear Logic Programming. Technical Report TR93-04, The University of Tokyo, Faculty of Science, Department of Information Science, 1993.
- Toru Sugimoto and Akinori Yonezawa. A preference-based theory of intention. Technical Report TR94-04, The University of Tokyo, Faculty of Science, Department of Information Science, 1993.

- Kenichi Asai, Satoshi Matsuoka, and Akinori Yonezawa. Roles of a Partial Evaluator for the Reflective Language Black. Technical Report TR94-11, The University of Tokyo, Faculty of Science, Department of Information Science, May 1994.
- Naoki Kobayashi, Motoki Nakade, and Akinori Yonezawa. Static Analysis on Communication for Asynchronous Concurrent Programming Languages. Technical Report TR95-04, The University of Tokyo, Faculty of Science, Department of Information Science, April 1995.
- Kenichi Asai, Hidehiko Masuhara, Satoshi Matsuoka, and Akinori Yonezawa. Partial Evaluator as a Compiler for Reflective Languages. Technical Report TR95-10, The University of Tokyo, Faculty of Science, Department of Information Science, December 1995.
- Kenjiro Taura and Akinori Yonezawa. Schematic: A Concurrent Object-Oriented Extension to Scheme. Technical Report TR95-11, The University of Tokyo, Faculty of Science, Department of Information Science, December 1995.
- Kenichi Asai, Hidehiko Masuhara, and Akinori Yonezawa. Partial Evaluation of Call-by-value lambda-calculus with Side-effects. Technical Report TR96-04, The University of Tokyo, Faculty of Science, Department of Information Science, November 1996.
- Yoshihiro Oyama, Kenjiro Taura, and Akinori Yonezawa. An Efficient Compilation Framework for Languages Based on a Concurrent Process Calculus. Technical Report TR97-07, The University of Tokyo, Faculty of Science, Department of Information Science, July 1997.

3.10 その他

- 岩元莞二, 杉本正勝, 奥乃博, 米澤明憲, 國藤進, 諏訪基. パネル討論会 人工知能とプログラミングの接点. *情報処理*, Vol. 26, No. 5, pp. 521–529, May 1985.
- Gul Agha, Samson Abramsky, Carl Hewitt, Robin Milner, Peter Wegner, and Akinori Yonezawa. Foundations of Concurrent Object-Oriented Programming (Panel). In *Proceedings of OOPSLA/ECOOP '90*, p. 100, Ottawa, Canada, October 1990. Also in SIGPLAN Notices 25(10).
- Gul Agha, Akinori Yonezawa, Peter Wegner, and Samson Abramsky. OOPSLA/ECOOP'90 Report, Panel: Foundations of Object-Based Concurrent Programming. In *Proceedings of the European Conference on Object-Oriented Programming Addendum*, pp. 9–14, Ottawa, Canada, 1990.
- Gul Agha, Akinori Yonezawa, Peter Wegner, and Samson Abramski. OOPSLA panel on object-based concurrent programming. *OOPS Messenger*, Vol. 2, No. 2, pp. 3–15, April 1991.
- Robert H. Halstead Jr., Takashi Chikayama, Richard P. Gabriel, David L. Waltz, and Akinori Yonezawa. Applications for Parallel Symbolic Computation (panel). In *Proceedings of US/Japan Workshop on Parallel and Symbolic Computing: Languages, Systems, and Applications, October, 1992*, Vol. 748, p. 417, Cambridge, MA, USA, 1993.
- Eiiti Wada and Akinori Yonezawa. Obituary: Professor Nobuo Yoneda (28 March 1930–22 April 1996). *Science of Computer Programming*, Vol. 27, No. 3, pp. 215–216, November 1996.
- 米澤明憲. インタラクティブ・エッセイ：私も日本のソフトをもっと強くしたい. *情報処理*, Vol. 39, No. 9, September 1998.
- 米澤明憲. C. A. R. Hoare 卿による京都賞ワークショップ講演について. *コンピュータソフトウェア*, Vol. 18, No. 4, p. 1, July 2001.
- 米澤明憲. 20世紀の名著論: John Backus: Can Programming Be Liberated from the von Neumann Style? A Functional Style and its Algebra of Programs. *情報処理*, Vol. 43, No. 9, p. 1017, September 2002.
- 大野豊, 片山卓也, 米澤明憲, 武市正人, 阿草清滋. 歴代理事長座談会. *コンピュータソフトウェア*, Vol. 20, No. 6, pp. 5–16, November 2003.

4 学位指導リスト

4.1 博士論文

1988 年度

- 大澤 一郎. 自然言語対話システムに関する基礎的研究.

1990 年度

- Takuo Watanabe. Object-Oriented Models for Reflection in Concurrent Systems. (渡部 卓雄. 並列システムにおける自己反映計算のモデルに関する研究.)

1992 年度

- Yuuji Ichisugi. A Reflective Object-Oriented Concurrent Language for Distributed Environments. (一杉 裕志. 分散環境におけるリフレクティブな並列オブジェクト指向言語.)

1993 年度

- Satoshi Matsuoka. Language Features for Extensibility and Re-use in Concurrent Object-Oriented Languages. (松岡 聡. 並列オブジェクト指向言語における拡張性と再利用のための言語機構.)
- Masahiro Yasugi. A Concurrent Object-Oriented Programming Language System for Highly Parallel Data-Driven Computers and its Applications. (八杉 昌宏. 高並列データ駆動計算機における並列オブジェクト指向プログラミング言語処理系とその応用プログラム.)

1994 年度

- Takeshi Fuchi. New Methods to Analyze Japanese Morphemes and Dependency Structure AND Formalization of Rules to Derive Implied Meanings. (淵 武志. 日本語形態素構文解析のための新手法および含意導出規則の定式化.)
- Toru Sugimoto. Formal Models of Dialogue Participants. (杉本 徹. 対話行為者の形式的モデルに関する研究.)
- Jacques Garrigue. Label-Selective Lambda-Calculi and Transformation Calculi. (ガリグ ジャック. 指定的ラムダ計算とトランスフォーメーション計算系.)

1995 年度

- Jeff McAffer. A meta-level architecture for prototyping object systems. (マカファー ジェフ. オブジェクトシステムのプロトタイピングのためのメタアーキテクチャ.)

1996 年度

- Naoki Kobayashi. Concurrent Linear Logic Programming. (小林 直樹. 並行線形論理プログラミング.)
- Takashi Miyata. A study on inference control in natural language processing. (宮田 高志. 自然言語処理における推論の制御に関する研究.)
- Naohito Sato. Modularity and Composability in an Object-Oriented Library Framework for Parallel and Distributed Computation. (佐藤 直人. 並列分散オブジェクト指向ライブラリフレームワークにおけるモジュラリティおよびコンポーザビリティ.)

1997 年度

- Kenichi Asai. The reflective language Black. (浅井 健一. 自己反映言語 Black.)
- Kenjiro Taura. Efficient and Reusable Implementation of Fine-Grain Multithreading and Garbage Collection on Distributed-Memory Parallel Computers. (田浦 健次朗. 分散記憶並列計算機のための効率的で再利用可能な細粒度マルチスレッディング及びゴミ集め.)
- Hiroshi Hosobe. Theoretical Properties and Efficient Satisfaction of Hierarchical Constraint Systems. (細部 博史. 階層制約系の理論的性質と効率的解消法.)

1998 年度

- Sachiko Kawachiya. Analyses and Reduction of Operational Overhead in Computer-Assisted Drawing. (河内谷 幸子. 計算機支援の描画における操作負荷の分析と削減.)

1999 年度

- Hidehiko Masuhara. Architecture Design and Compilation Techniques Using Partial Evaluation in Reflective Concurrent Object-Oriented Languages. (増原 英彦. 並行オブジェクト指向言語における自己反映アーキテクチャの設計と部分計算を用いたコンパイル方法.)
- Tatsuro Sekiguchi. A Study on Mobile Language Systems. (関口 龍郎. モバイル言語システムに関する研究.)

2000 年度

- Haruo Hosoya. Regular Expression Types for XML. (細谷 晴夫. XML のための正規表現型.)
- Yoshihiro Oyama. Achieving High Performance for Parallel Programs that Contain Unscalable Modules. (大山 恵弘. スケーラブルでないモジュールを含む並列プログラムにおける高性能の達成.)

2001 年度

- Toshio Endo. Scalable Dynamic Memory Management Module on Shared Memory Multiprocessors. (遠藤 敏夫. 共有メモリ並列計算機上のスケーラブルな動的メモリ管理モジュール.)

2004 年度

- Yutaka Oiwa. Implementation of a Fail-Safe ANSI C Compiler. (大岩 寛. 安全な ANSI C コンパイラの実装手法.)
- Eric Y. Chen. Defending against Distributed Denial of Service Attacks (分散サービス停止攻撃の対策方式).
- Reynald Affeldt. Verification of Concurrent Programs using Proof Assistants (定理証明器に基づく並行プログラムの検証).

2005 年度

- Toshiyuki Maeda. Writing an Operating System with a Strictly Typed Assembly Language. (前田 俊行. 厳密に型付けされたアセンブリ言語を用いたオペレーティングシステムの記述.)
- Kenji Kaneda. Middleware Systems for Enabling users to Adapt to Dynamic Changes in Execution Environments. (金田 憲二. 実行環境の動的な変化にユーザが適応することを可能にするためのミドルウェアシステム.)

2007 年度

- Kohei Suenaga. Type Systems for Formal Verification of Concurrent Programs. (末永 幸平. 並行プログラムの形式的検証のための型システム.)
- Marti Nicolas. Formal Verification of Low-level Software (低レベルソフトウェアの形式的な証明).

2008 年度

- Hiroshi Unno. Dependent Type Inference for Program Verification. (海野 広志. プログラム検証のための依存型推論.)

2009 年度

- Koichi Onoue. VMM-based Systems for Enhancing Application Security. (尾上 浩一. VMM を利用した アプリケーションプログラムの安全性を向上させるシステム.)

2010 年度 (予定)

- Masaaki Shimizu. Operating System Structures for High Performance Computer Clusters. (清水 正明. 高性能計算機クラスタのためのオペレーティングシステム構成法.)
- Dun Nan (頓 楠). Rapidly Deployable, Scalable, and High-Performance Distributed File System for Data-Intensive Distributed Computing.

4.2 修士論文

1984 年度

- 三ツ井 欽一. An Object Oriented Approach for Natural Language Comprehension.

1987 年度

- 小泉 博嗣. ダイアグラムを用いたプログラミング.
- 高田 敏弘. 並列オブジェクト指向言語の分散環境における実現.
- 本田 康晃. Debugging Concurrent Systems Based on Object Groups.
- 松井 知子. ニューラルネットによる自然言語処理の研究.
- 渡部 卓雄. Reflection in Object-Oriented Concurrent Systems.

1988 年度

- 塚田 元. 談話理解における名詞句の扱いについて.

1989 年度

- 一杉 裕志. 並列オブジェクト指向言語の分散環境での実現

1991 年度

- Kenichi Asai. Model Checking of Finite State CSP. (浅井 健一. 有限状態 CSP のモデル検査.)
- Tsuyoshi Murata. A Framework for Describing Mental States and their Dynamic Changes of Dialog Participants. (村田 剛志. 対話参加者の心的記述とその動的変化に関する枠組.)

1992 年度

- Naoki Kobayashi. Asynchronous Communication Model based on Linear Logic. (小林 直樹. 線形論理に基づく非同期通信モデル.)
- Takashi Miyata. Dialogue System Based on Dynamical Constraint Programming. (宮田 高志. 力学制約プログラミングに基づいた対話システム.)
- Shin Takahashi. A Framework for Constructing Animations via Declarative Mapping Rules. (高橋 伸. 宣言的対応規則によるアニメーション作成のための枠組み.)
- Yoshisato Takeda. RNA Secondary Structure Prediction Using Highly Parallel Computer. (竹田 義聡. 高並列計算機による RNA の 2 次構造予測)
- Shin'ichi Furuso. Parallel Conservative Garbage Collection with "Almost Synchronization-free" Fast Object Allocation. (古荘 進一. 同期のほぼ不要にオブジェクト割り付けの可能な Conservative Garbage Collection.)

1993 年度

- Kenjiro Taura. Design and Implementation of Concurrent Object-Oriented Programming Languages on Stock Multicomputers. (田浦 健次郎. 並列オブジェクト指向言語のマルチコンピュータ上における設計および実装.)
- Hidehiko Masuhara. Study on a Reflective Architecture to Provide Efficient Dynamic Resource Management for Highly-Parallel Object-Oriented Applications. (増原 英彦. 高並列オブジェクト指向アプリケーションのための効率のよい動的資源管理方式を提供する自己反映アーキテクチャの研究.)
- Kentaro Torisawa. Lexicon Acquisition in HPSG-based Grammars. (鳥澤 健太郎. HPSG をベースとする文法における辞書項目の獲得.)
- Naohito Omori. An Application of Parallel Object-Oriented Methodology to Molecular Dynamics Simulation. (大森 直人. 並列オブジェクト指向プログラミングの分子動力学計算への応用.)
- Masaaki Nagatsuka. Compiler Design and Implementation Techniques for Parallel Constraint Logic Programming Languages. (長塚 雅明. 並列制約論理型言語のコンパイラ設計と実装技法.)
- Ken Miyashita. Interactive Generation of Graphical User Interfaces by Multiple Visual Examples. (宮下 健. 複数の視覚的例による直接操作インターフェースの対話的実現.)

1994 年度

- Tomio Kamada. A Study on Debugging Schemes in Concurrent Programs on Massively Parallel Processors. (鎌田 十三郎. 超並列計算機上の並列プログラム用デバッグ方式の研究.)
- Kazuhiro Konno. An Extension to a Parallel Constraint Logic Programming Language For Applications in Optimization Problems. (今野 和浩. 最適化問題への応用のための並列制約論理型言語の拡張.)
- Tatsuro Sekiguchi. A Study on Structure of Concurrent Object-Oriented Languages. (関口 龍郎. 並列オブジェクト指向言語の構造に関する研究.)
- Hiroshi Hosobe. Efficient Satisfaction of Constraint Hierarchies. (細部 博史. 制約階層の効率的な解消法.)

1995 年度

- Toshihiro Shimizu. A Study on Implementation of a Typed Concurrent Programming Language on Distributed Machines. (清水 智弘. 型付並列言語の分散メモリマシンへの実装.)

- Akihiro Nakaya. RNA Secondary Structure Prediction and Visualization Using Highly Parallel Computers. (中谷 明弘. 高並列計算機を用いた RNA の 2 次構造予測と視覚化.)

1996 年度

- Haruo Hosoya. Type inference garbage collection and its parallelization. (細谷 晴夫. 型推論ガベージコレクションとその並列アルゴリズム.)
- Atsushi Igarashi. Type-based Analysis of Usage of Values for Concurrent Programming Languages. (五十嵐 淳. 並列言語における型システムによる値の使用回数の解析.)

1997 年度

- Toshio Endo. A Scalable Mark-Sweep Garbage Collector on Large-Scale Shared-Memory Machines. (遠藤 敏夫. 大規模共有メモリマシン上のスケーラブルなマークスイープ法ガベージコレクタ.)
- Yoshihiro Oyama. An Efficient Compilation Framework for Parallel Programming Languages Based on a Concurrent Process Calculus. (大山 恵弘. 並行計算に基づく並列プログラミング言語のための効率的なコンパイルの枠組み.)

1998 年度

- Norifumi Gotoh. Software-based Fine-grained Memory Protection Supporting Dynamic Changes of Attributes. (後藤 礼史. 保護属性の動的変更が可能なソフトウェアによる細粒度メモリ保護.)
- Kunio Tabata. Lazy Task Creation on Shared Memory Machines for C Programs. (田端 邦男. 共有メモリ並列計算機上における C プログラムのための Lazy Task Creation.)
- Shuntaro Hitomi. Research on Effects of Mobile Object Programming with Describing Distributed Application. (人見 俊太郎. 移動オブジェクトプログラミングの分散処理記述による有効性の研究.)
- Hirotaka Yamamoto. Implementation of an Efficient and Reusable Distributed Garbage Collector and Evaluation of Its Performance. (山本 泰宇. 効率的で再利用可能な分散 GC ライブラリの設計と性能評価.)

1999 年度

- Takeo Imai. Dynamic Access Control of Mobile Objects by Switching Name Spaces. (今井 健男. 名前空間の切り替えによる移動オブジェクトの動的アクセス制御.)

- Sumio Toyama. Lazy Task Creation with Stack Swapping to Improve Memory Locality. (外山 純生. メモリ局所性を高めるスタック交換方式を導入した遅延タスク生成.)
- Nayuta Taga. An extension of C++ that supports thread migration with little loss of normal execution efficiency. (多賀 奈由太. 通常の実行効率を損わないスレッドマイグレーションが可能な C++.)

2000 年度

- Yutaka Oiwa. A Java-based Language With Type-safe Dynamic Code Generation. (大岩 寛. 型安全な動的コード生成をサポートする Java の言語拡張.)
- Shuhei Ohkado. Evaluation of Expressiveness and Performance of OpenMP Using a Suite of SPLASH2 Applications. (大角 周平. SPLASH2 アプリケーション集を用いた OpenMP の記述性と性能の評価.)
- Hideki Kariya. Efficient Type Inference for the Quasi-linear type system. (假家 英樹. 擬似線形型システムの効率的な型推論.)
- Yoshizumi Tanaka. Implementation and Evaluation of Nested Parallelism in OpenMP. (田中 義純. OpenMP における入れ子並列性の実装と評価.)
- Daisuke Hoshina. A Study on a Typed Process Calculus for Access Control in Distributed Computation. (星名 大輔. 分散計算におけるアクセスコントロールのための型付きプロセス計算の研究.)

2001 年度

- Reynald Affeldt. Supporting Object-Oriented Features in Run-time Bytecode Specialization. (実行時バイトコード特化のオブジェクト指向言語への拡張.)
- Yohei Ueda. A Self-organizing Hierarchical Network for Global Parallel Computing. (上田 陽平. 広域分散並列計算のための自己編成可能な階層的ネットワーク.)
- Takahiro Sakamoto. Dynamic Thread Mapping for Distributed Data Intensive Applications. (坂本 崇裕. 分散データ intensive なアプリケーションのための動的スレッド配置.)
- Nobuyuki Hamanaka. A Type System for Object Usage Analysis in Java Bytecode. (浜中 信行. Java バイトコードにおけるオブジェクトの使用解析のための型システム.)
- Soonsang Hong. Distributed web crawling with dynamically joining/leaving resources. (洪 淳祥. 動的に増減する資源を利用した広域分散 web crawling.)
- Toshiyuki Maeda. Safe Execution of User Programs in Kernel Mode using Typed Assembly Language. (前田 俊行. 型付きアセンブリ言語を用いてユーザプログラムをカーネルモードで実行する方法.)

- Hiroshi Yamauchi. Designing a Virtual Machine Independent Compiler Interface. (山内 裕史. 仮想マシン独立なコンパイラインタフェースの設計.)

2002 年度

- Kenji Kaneda. Implementing a Runtime System for Parallel and Distributed Computation on the Internet and Its Applications. (金田 憲二. インターネット上の並列分散計算のための実行時システムとそのアプリケーションの実装.)
- Yoshinori Kobayashi. An Efficient Garbage Collector in the Presence of Ambiguous References. (小林 義徳. 曖昧なポインタの存在下での効率的なゴミ集め.)
- Naoshi Tabuchi. Regular Expression Types for Strings in a Text Processing Language. (田淵 直. テキスト処理言語における文字列のための正規表現型.)
- Yuuta Hayami. Java Bytecode Transformation for Fine Grain CPU Resource Management. (速水 雄太. Java バイトコード変換による細粒度 CPU 資源管理.)
- Tatsuo Mizukami. Resource Usage Analysis with References. (水上 達夫. 参照を扱った資源使用解析.)

2003 年度

- Masashi Seiki. Model Checking for Computer Game Scenarios. (清木 昌. モデル検査理論のゲームシナリオへの適用.)
- Akihito Nagata. Region-Based Memory Management for Dynamic Typed Language. (永田 章人. 動的型付き言語のためのリージョン推論に基づくメモリ管理.)
- Takashi Masuyama. Designing and Implementing a Distributed Component Language with Transparent Mobility. (増山 隆. 透明な移動が可能な分散コンポーネント言語の設計と実装.)

2004 年度

- Yuichi Oneda. A Bytecode Translation for Pointcut Description Based on Data Dependency. (大根田 裕一. データ間依存性に基づくポイントカット記述のためのバイトコード変換.)
- Koichi Onoue. Design and Implementation of a Mobile Computing System Based on a CPU Emulator. (尾上 浩一. CPU エミュレータを用いた移動計算システムの設計と実装.)
- Kohei Suenaga. Translation of Tree-processing Programs into Stream-processing Programs Based on Ordered Linear Types. (末永 幸平. 順序付き線形型に基づく木構造処理プログラムからストリーム処理プログラムへの変換.)
- Hideaki Tatsuzawa. An Aspect-oriented Language for Functional Programming. (立沢 秀晃. アスペクト指向のための関数型プログラミング言語.)

- Akihito Tokuda. An Integrated Approach to Assessing the Reliability of Large-Scale Protein-Protein Interaction Data. (徳田 顕人. 大規模蛋白質間相互作用データの統合的な信頼性評価法.)
- Hirofumi Fujimoto. A Fault-tolerant, Autonomic, Adaptive Search Engine. (藤本 浩史. 耐故障性を備えた自律適応型検索エンジン.)

2005 年度

- Hiroshi Unno. Combining Type-Based Analysis and Model Checking for Finding Counterexamples against Non-Interference. (海野 広志. 型解析とモデル検査を組み合わせた非干渉性の反例発見法.)
- Yusuke Endo. Continuation Join Point. (遠藤 侑介. 継続ジョインポイント.)
- Yousuke Yokoyama. Application-Specific Virtual Environments for Migration and Checkpointing. (横山 陽介. アプリケーションに特化した計算移動やチェックポイントングのための仮想環境.)
- Toshihiro Yoshino. A Framework Using a Common Language to Build Program Verifiers for Low-Level Languages. (吉野 寿宏. 低級言語のプログラム検証器を構成するための共通言語を用いたフレームワーク.)

2006 年度

- Hideaki Sato. Fast and Precise Code Clone Reduction by Incremental Analysis. (佐藤 秀明. 差分解析による高速かつ正確なコードクローンの解消.)
- Daisuke Shimamoto. Detecting Anomalies on Windows by Monitoring System Services. (島本 大輔. System Service 監視による Windows 向け異常検知.)
- Yoshiki Nishikawa. Distributed System Test Bed by Virtualization of the View for OS Resources. (西川 賀樹. OS 資源ビューの仮想化を用いた分散システムテストベッド.)
- Dun Nan (頓 楠). Design and Implementation of Locality-Aware Distributed File System for Wide-Area Grid Environments.

2007 年度

- Takahiro Yamazaki. Reactive Model Checking for Game Scenarios. (山崎 孝裕. ゲームシナリオのための反応性の良いモデル検査.)
- Yasushi Oda. Design and Implementation of a Strongly Typed Intermediate Language for GCC. (小田 泰. GCC のための強く型付けされた中間言語の設計と実装)
- Haruki Sato. Extending Alias Types with Separating Implication. (佐藤 春旗. Separating Implication による Alias Types の拡張.)

- Hideyuki Tanaka. Automatic Parallelization of Purely Functional Programs with Estimation of Function Evaluation Cost. (田中 英行. 関数評価のコストの推定を用いた純粋関数型言語の自動並列化.)
- Ryoza Yamashita. Secure Information Flow for Resources. (山下 諒蔵. リソースのための情報流解析.)

2008 年度

- Hiroyuki Osumi. A Distributed Storage System for Virtual Disk Management. (大住 裕之. 仮想ディスク管理のための分散ストレージシステム.)
- Takahiro Kosakai. A Typed Assembly Language with Reference Counting. (小酒井 隆広. 参照カウントを用いる型付きアセンブリ言語.)
- Wataru Takaki. A Merging Facility for a Tree-Structured Versioning File System. (高木 渉. ツリー構造バージョン管理ファイルシステムのためのマージ機構.)

2009 年度

- Daisuke Iizuka. Stack Types with Bounded Existential Quantification. (飯塚 大輔. 境界つき存在量化子を用いたスタック型.)
- Junya Sawazaki. A Hybrid Virtual Machine Monitor for Implementing Flexible and High-Performance Security Mechanisms. (澤崎 純也. 柔軟かつ高性能なセキュリティ機構を実装するためのハイブリッド仮想マシンモニタ.)
- Takahiro Nojiri. Parallelization of Safety Verification in SPIN Model Checker. (野尻 隆宏. モデル検査器 SPIN における安全性検証の並列化.)

2010 年度 (予定)

- Tomohiro Suzuki. Programming Language Support for Multiple Mobile Ad Hoc Networks. (鈴木 友博. 複数のモバイルアドホックネットワークのためのプログラミング言語サポート.)
- Yuki Watanabe. Study on Proof of Type Preservation in CPS Transformation. (渡邊 裕貴. CPS 変換における型保存の証明に関する研究.)

4.3 卒業論文

1989 年度

- Kenichi Asai. Multilisp Implementation. (浅井 健一. マルチリスプの実現.)
- Hideaki Kuwada. Fast Implementation of Production Systems. (桑田 英明. プロダクションシステムの高高速化.)
- Tsuyoshi Murata. FDS-A Flexible Dialog System. (村田 剛志. 柔軟な対話システム FDS.)
- Susumu Hasegawa. A Study of Hopfield Network. (長谷川 進. Hopfield Network に関する研究.)

1990 年度

- Naoki Kobayashi. Control of Parallel Computation in Constraint Logic Programming. (小林 直樹. 制約論理プログラムにおける並列計算の制御.)
- Shin Takahashi. User Interface on a Model of Bi-directional Translation Between Picture and Data. (高橋 伸. 絵-データ間の双方向変換に関するモデルに基づいたユーザインターフェース.)
- Yoshisato Takeda. Dynamic and Reflective Scheduling in Concurrent Object-Oriented Programming Language. (竹田 義聡. 並列オブジェクト指向言語における動的で自己反映的なスケジューリング.)
- Tsuyoshi Hayashi. A Distributed Problem Solving Platform with Distributed Blackboard Architecture. (林 毅. 分散ブラックボード・アーキテクチャに基づいた分散問題解決のプラットフォーム.)
- Shin'ichi Furuso. Garbage Collection in A Concurrent Architecture. (古荘 進一. 並列アーキテクチャ上の Garbage Collection.)
- Takashi Miyata. Domain Independent Dialog System. (宮田 高志. 獲得すべきデータの仕様を独立させた対話システム.)

1991 年度

- Daisuke Suzuki. Inductive Inference under Algebraic Specifications. (鈴木 大介. 代数的仕様の下での帰納的推論について.)
- Kenjiro Taura. A Study of an Implementation of an Object-Oriented Concurrent Language on Distributed Memory Parallel Architecture. (田浦 健次郎. マルチコンピュータ上の並列オブジェクト指向言語の高効率な実装に関する研究.)
- Kentaro Torisawa. Acquiring Discourse Domain Knowledge with a Dialog System. (鳥澤 健太郎. 対話領域に関する知識の対話システムによる獲得)

- Masaaki Nagatsuka. The Implementation of Constraint Logic Programming on Parallel Computer. (長塚 雅明. 制約論理型言語の並列計算機上への実装.)
- Hidehiko Masuhara. Study on Implementation of an Object-Oriented Concurrent Reflective Language. (増原 英彦. 自己反映計算の機能を持つ並列オブジェクト指向言語の実装に関する研究.)
- Ken Miyashita. Programming by Visual Example. (宮下 健. 視覚的例によるプログラミング.)

1992 年度

- Kazuhiro Konno. An Efficient Implementation of a Parallel Constraint Logic Programming Language and its Applications. (今野 和浩. 並列制約論理型言語の効率的な実装と応用.)
- Tomio Kamada. An Algorithm of Distributed Garbage Collections on a Multicomputer and its Performance Evaluation. (鎌田 十三郎. 並列計算機上の分散ガーベージ・コレクションのアルゴリズムとその性能評価.)
- Hiroshi Hosobe. A Constraint Solving Method for Real-Time Interaction in User Interfaces. (細部 博史. ユーザインターフェースで実時間のインタラクションを実現する制約解消法.)
- Kengpang Lee. An Object-Oriented Approach to Parsing for Unification-based Grammars. (李 敬邦. オブジェクト指向方式による単一化文化を用いたパーシング.)
- Tatsurou Sekiguchi. Construction of Type Inference Systems for Concurrent Object-Oriented Languages. (関口 龍郎. 並列オブジェクト指向言語のための型推論体系の構築.)

1993 年度

- Akihiro Nakaya. Type Inference for Polymorphic Reference with Subtyping. (中谷 明弘. 部分型の入った多相参照のための型推論.)
- Toshihiro Shimizu. A study on an implementation of HACL compiler on multicomputers. (清水 智弘. HACL コンパイラのマルチコンピュータ上での実装技術の研究.)

1994 年度

- Atsushi Igarashi. Study on Mechanisms For Multi-Object Synchronization and Their Implementation. (五十嵐 淳. オブジェクト間同期機構とその実装に関する研究.)
- Motoki Nakade. Static Analysis on Communication for Asynchronous Concurrent Programming Languages. (中出 元樹. 非同期並行言語における通信の静的解析.)

- Haruo Hosoya. Control Constructs in Concurrent Object-Oriented Languages. (細谷 晴夫. 並列オブジェクト指向言語における制御構造.)
- Daisuke Bekki. Dialogue processing with Mental World Structure. (戸次 大介. 心的世界構造を用いた対話処理.)
- Satoshi Moriwaki. Construction of a Japanese-text Generation System With Semantic-Head-Driven Generation. (森脇 敏. 意味主辞駆動生成アルゴリズムを用いた日本語文生成システムの構築.)
- Chikashi Nobata. Automatic Acquisition of The Cost of Connectivity in A Japanese Morphological Analyzer. (野畑 周. 日本語形態素解析システムにおける接続コストの自動獲得.)

1995 年度

- Toshio Endo. A Methodology for Constructing a Portable Garbage Collector on Parallel Machines. (遠藤 敏夫. 並列マシンにおける Portable な Garbage Collector の実装法.)
- Yoshihiro Oyama. A General Framework for Compiling Fine-grain Threads in Concurrent Object-Oriented Languages. (大山 恵弘. 並列オブジェクト指向言語における細粒度スレッドコンパイルのための一般的枠組み.)

1996 年度

- Norifumi Gotoh. Improving execution efficiency of Tree-Structure-Based Parallel Programs. (後藤 礼史. 木構造を扱う並列プログラムの効率化.)
- Shuntaro Hitomi. Mobile Emacs Lisp: An Extension to Emacs Lisp for Efficient Editing on Wide-Area Network. (人見 俊太郎. Mobile Emacs Lisp: 広域ネットワーク上の効率的な編集作業のための Emacs Lisp の拡張.)
- Hirotaka Yamamoto. A Research on Execution Performance of Parallel Programs on a Workstation Cluster. (山本 泰宇. 並列プログラムのワークステーションクラスタ上における実行効率の研究.)

1997 年度

- Sumio Toyama. Implementation and Evaluation of Concurrent Object in Schematic. (外山 純生. Schematic における並列オブジェクトの実装と評価.)
- Nayuta Taga. Implementation of Mobile Scheme. (多賀 奈由太. モバイル Scheme の実装.)

1998 年度

- Yutaka Oiwa. Extending Java Virtual Machine to Improve Performance of Dynamically-Typed Languages. (大岩 寛. 動的型付き言語の効率向上のための Java 仮想マシンの拡張.)
- Shuhei Ohkado. Adding Fine-Grain Threading to Python Using StackThreads/MP. (大角 周平. StackThreads/MP による Python への細粒度スレッドの追加.)
- Yoshizumi Tanaka. Copying Garbage Collection in the Presence of Uncertain Pointers. (田中 義純. 不確かなポインタの存在下でのコピーイングガーベジコレクション.)

1999 年度

- Yohei Ueda. Single Memory Image for Heterogeneous Distributed Computing: Its Implementation and Evaluation. (上田 陽平. 異種計算機間の分散計算のための単一メモリイメージの実現と性能評価.)
- Takahiro Sakamoto. Java Bytecode Translation for Transparent Migration. (坂本 崇裕. 透明なマイグレーションのための Java バイトコード変換.)
- Nobuyuki Hamanaka. An Implementation of Distributed Cyclic Garbage Collection for Java RMI. (浜中 信行. Java RMI での分散環状ゴミ集めの実装.)
- Hiroshi Yamauchi. Automatic Library Translation for Host-local Resources in Languages with Object Mobility. (山内 裕史. オブジェクト移動性を持った言語におけるホストローカルなリソースのための自動ライブラリ変換.)

2000 年度

- Futoshi Iwama. A New Type System for JVM Lock Primitive. (岩間 太. JVM におけるロックの整合性検証のための 新しい型システム.)
- Kenji Kaneda. Virtual Private Grid (VPG): A Command Shell for Utilizing Remote Machines Efficiently. (金田 憲二. Virtual Private Grid (VPG): 遠隔計算機を効率的に利用するシェル.)
- Yuuta Hayami. Java Bytecode Transformation for Fine Grain CPU Resource Management. (速水 雄太. Java バイトコード変換による細粒度 CPU 資源管理.)
- Tatsuo Mizukami. Design and Implementation of Storage Server for Mobile Clients. (水上 達夫. モバイルクライアントのためのストレージサーバーのデザインと実装.)

2001 年度

- Nguyen Viet Ha. Parallelizing Programs Using Access Traces. (グエン ヴェトハ. アクセス履歴によるプログラムの並列化.)

- Masashi Seiki. A Scenario Description Language Based on a State Transition Model and its Automatic Consistency Checking. (清木 昌. 状態遷移モデルに基づいたシナリオ記述言語の設計と同言語上での自動検証.)
- Akihito Nagata. Scheme-to-Java Translator with Soft Typing. (永田 章人. Soft Typing を利用した Scheme-to-Java トランスレータ.)
- Takashi Masuyama. An Approach to Separate Compilation of C++ Templates. (増山 隆. C++のテンプレート関数を分割コンパイル可能にするためのアプローチ.)

2002 年度

- Kohei Suenaga. The Interface Definition Language for Fail-Safe C. (末永 幸平. Fail-Safe C のためのインターフェイス定義言語.)
- Yuichi Oneda. CPS Conversion of Procedural Language with Exception Handling and Its Formalization. (大根田 裕一. 例外処理機構を備えた手続き型言語の CPS 変換とその定式化.)
- Takuo Suzuki. Packrat Parsing with State. (鈴木 拓男. 状態つき Packrat Parsing.)
- Hideaki Tatsuzawa. Translating Security Protocols from Informal Notation into Spi Calculus. (立沢 秀晃. セキュリティプロトコルの略式表現から spi calculus への変換.)

2003 年度

- Yusuke Endo. Aspect-Oriented Virtual Machine for Supporting Dynamic Weaving and High-Level Features. (遠藤 侑介. 高級な言語機能を持ちアスペクトを動的に適用する仮想機械.)
- Shinya Kawanaka. Region Representation Inference for Dynamically-Typed Language. (川中 真耶. 動的型付け言語の為のリージョン表現推論.)
- Yosuke Yokoyama. A Toolkit for Developing Extensible and Portable Checkpoint Systems. (横山 陽介. 拡張性と互換性を持つチェックポイントシステムの構築を容易にする方法.)
- Toshihiro Yoshino. Design and Implementation of a Self-Repairing Reference Monitor. (吉野 寿宏. 自己修復型リファレンスモニタの設計と実装.)

2004 年度

- Hideaki Sato. Effective Software Obfuscation by Mixing Instructions and Data. (佐藤 秀明. 命令とデータの混合による効果的なソフトウェアの難読化.)
- Daisuke Shimamoto. Detecting Intrusions on Windows Operating Systems by Monitoring System Services. (島本 大輔. System Service の監視による Windows 用侵入検知システム.)

- Takahiro Yamazaki. Reliable Peer-to-Peer Framework for Massively Multiplayer Online Games. (山崎 孝裕. 大規模多人数オンラインゲームのための信頼性のある Peer-to-Peer の枠組み.)

2005 年度

- Hiroyuki Osumi. A Library for Replay-based Recovery. (大住 裕之. リプレイに基づくリカバリのためのライブラリ.)
- Haruki Sato. Implementation of Programming Language based on Alias Types and Measurement of Its Ability. (佐藤 春旗. Alias Types に基づくプログラミング言語の実装とその評価.)
- Ryozo Yamashita. Implementation of Information Flow Analysis using Dynamic Type Checking. (山下 諒蔵. 動的な型検査を用いた情報流解析の実装.)

2006 年度

- Daisuke Iizuka. Typed Assembly Language for Conditional Execution. (飯塚 大輔. 条件実行に対応する型付きアセンブリ言語.)
- Takahiro Kosakai. Compiling C Programs into a Type-safe Assembly Language. (小酒井 隆広. C 言語から型安全なアセンブリ言語へのコンパイル.)
- Satoshi Goto. Examination of Application of Model Checking for Automatic Verification of Safety of Memory Operation. (後藤 哲志. モデル検査の応用によるメモリ操作の安全性を自動検証する手法に関する研究.)

2007 年度

- Sukehide Ushioda. Dependent Types in Imperative Programming Language. (潮田 資秀. 命令型プログラミング言語における依存型.)
- Junya Sawazaki. Detecting Creation of Processes and Execution of Applications without Modifying Guest OS Kernel on Virtual Machine Monitor. (澤崎 純也. OS カーネルを修正することなく仮想マシンモニタ上でのプロセス生成とアプリケーション実行を検出する手法.)
- Takahiro Nojiri. Array Bounds Inference with a Model Checker for a Dependent Type System. (野尻 隆宏. モデル検査器を用いた依存型システムのための配列境界の推論.)
- Masayuki Murata. Constraint Solver for Bounded Integer with Correctness Proof. (村田 雅之. 正しさの証明が与えられた有限範囲の整数に対する制約解消器.)

2008 年度

- Keiichiro Uemura. Realizing Fairness in StackThreads/MP Scheme. (植村 圭一朗. StackThreads/MP への公平性の導入.)

- Tomohiro Suzuki. A Honey-pot System Solving Dilemma Between Freedom and Restriction. (鈴木 友博. 攻撃者に与える自由と制約のジレンマを解消するハニーポットシステム.)
- Yuki Watanabe. Extending Hoare Type Theory for Arrays. (渡邊 裕貴. 配列のためのホーア型理論の拡張.)

2009 年度

- Takuro Ikejiri. Encapsulation in Prototype-Based Object-Oriented Language using Ownership Types. (池尻 拓朗. Ownership Types を用いたプロトタイプ・ベースオブジェクト指向言語のカプセル化.)

2010 年度 (予定)

- Junya Arai. Lightweight Implementation of Profile-Driven Implicit Parallelization for Haskell. (新井 淳也. Haskell のためのプロファイル駆動暗黙的並列化の簡潔な実装.)
- Yuto Takei. Analysis and Sugesstion on Security Requirements for Software Development. (竹井 悠人. ソフトウェア開発におけるセキュリティ要件の分析と提案.)
- Masato Tanaka. Improving BEEP Approach for Preventing XSS Attacks and Reducing Its Overhead. (田中 雅人. XSS 攻撃の防御のための BEEP 手法の改善とそのオーバーヘッドの削減.)

米澤研究室（東京大学・東京工業大学）関係者・OB 名簿

学生 青柳 滋巳, 秋葉 友良, 秋山 茂樹, 浅井 健一, 新井 淳也, 飯塚 大輔, 五十嵐 淳, 池尻 拓朗, 一杉 裕志, 今井 健男, 岩間 太, 上田 晴康, 上田 陽平, 植村 圭一郎, 潮田 資秀, 海野 広志, 遠藤 敏夫, 遠藤 侑介, 大岩 寛, 大角 周平, 大澤 一郎, 大住 裕之, 大野 忠久, 大山 恵弘, 大根田 裕一, 小田 泰, 尾上 浩一, 柿元 亮一, 加藤 優, 金田 憲二, 金田 重治, 鎌田 十三郎, 鴨志田 良和, Jacques Garrigue (ジャック ガリグ), 假家 英樹, 川中 真耶, 木内 康彦, 岸本 章宏, 河内谷 幸子, 桑田 英明, Nguyen Viet Ha (グエン ヴェトハ), 黒須 博文, 小泉 博嗣, 小酒井 隆広, 後藤 哲志, 後藤 礼史, 小林 直樹, 小林 義徳, 今野 和浩, 斉藤 新, 酒井 伸啓, 坂本 崇裕, 佐口 泰之, 佐瀬 幹哉, 佐藤 一郎, 佐藤 直人, 佐藤 春旗, 佐藤 秀明, 佐野 勝也, 澤崎 純也, Jean-Marc Jezequel (ジャン マーク ジェゼクル), 塩谷 沢生, 島本 大輔, 清水 智弘, 清水 正明, 柴山 悦哉, 末永 幸平, 杉田 祐也, 杉本 徹, 鈴木 大介, 鈴木 拓男, 鈴木 友博, 住井 英二郎, 清木 昌, 関口 龍郎, 田浦 健次朗, 高木 渉, 高田 敏弘, 高橋 伸, 高橋 俊行, 高橋 秀之, 多賀 奈由太, 竹井 悠人, 竹田 義聡, 武山 誠, 立沢 秀晃, 田中 英行, 田中 雅人, 田中 義純, 田端 邦男, 田淵 直, 玉越 大輝, Eric Y. Chen (エリック チェン), 塚田 元, 鳥澤 健太郎, 徳田 顕人, 外山 滋, 外山 純生, 頓 楠, 永田 章人, 長塚 雅明, 中尾 晴彦, 中川 貴之, 中出 元樹, 中村 敏幸, 中谷 明弘, 永野 稔夫, Marti Nicolas (マーティ ニコラ), 西川 浩, 西川 賀樹, 野里 貴仁, 野尻 隆宏, 野畑 周, 袴谷 忠靖, 萩原 馨, 浜中 信行, 林 毅, 速水 雄太, 原 正巳, 人見 俊太郎, 淵 武志, 古荘 進一, 藤本 浩史, 藤原 頼晶, 星名 大輔, 戸次 大介, 細部 博史, 細谷 晴夫, 洪 淳祥, 本田 康晃, 前田 俊行, Jeff McAffer (ジェフ マカファー), 増原 英彦, 増山 隆, 松井 知子, 松岡 聡, 松島 栄樹, 松田 裕幸, 丸山 友朗, 水上 達夫, 三ツ井 欽一, 宮下 健, 宮田 高志, 武舎 広幸, 村田 剛志, 村田 雅之, 森脇 敏, 八杉 昌宏, 山内 裕史, 山崎 孝裕, 山下 諒蔵, 山本 泰宇, 横山 陽介, 吉野 寿宏, 李 敬邦, Reynald Affeldt (レナルド アフェルト), 渡部 卓雄, 渡邊 裕貴

研究員 Yariv Aridor (ヤリブ アリドール), 高 耀清, Rajeev Surati (ラジャブ スラティ), Laurent Thomas (ローラン トマ), Kresten Krab Thorup (クレストン クラブ トロプ), David Nowak (ダヴィッド ノヴァック), 橋本 政朋, 古瀬 淳, Jean-Pierre Briot (ジャン ピエル ブリオ), Frederic Peschanski (フレデリック ペシャンスキ), 松田 元彦, Khaled Ragab (ハレド ラガブ)

スタッフ 大津賀 純子, 岡田 桂子, 小林 庸子, 斎藤文江, 坂西 由美子, 柴田 春子, 鈴木 由美子, 松本 典子

 パイオニア

私のソフトウェア研究

米澤 明憲

1 はじめに

このたび、個人の学術的な貢献を顕彰するフェロー制度が日本ソフトウェア科学会に創設され、その第一回の受賞者の一人としてお選びいただいたことは、誠に身に余る光栄と存じます。制度の創設・選考に携われた先生方のご尽力に、深い敬意と感謝の意を心から表したいと存じます。

本学会の理事の方からいただいたメールの中には、私が以前に「コンピュータソフトウェア」に書いた2つの論文が挙げられており、好きな方を1つ選べばそれを近いうちに発行される同誌に再掲して下さることであった。そして、その再掲される論文にあまりページ数にこだわらずに序文をつけてよい、またその序文のタイトルを何とするつもりか至急知らせよ、とあった（もちろん締切はタイトであるとも）。

苦し紛れにここにあるようなタイトルで返答してしまったので、本年57歳で（そのままいれば）あと7、8年はお勤めできる現在の所属大学を「定年」で退職する際の最終講義のタイトルを、あたかも早めに口走ってしまったような気がして、いまだにパツが悪い。しかし、あまり面白いことも書けないので、厚顔無恥なる年長者の昔話と意固地な信条の一端にお付き合いいただければ、誠に有難い次第である。長くなるかもしれないので、その信条の一部については、あらかじめ、箇条書的に要約しておく（もちろん、理想

を述べているに過ぎず、私がすべてを実践できたものではない）。

1. 奥の深そう (pregnant) な着想を見つけて、長期間研究する。
2. 研究の成果に対して明確な目的意識をもって研究する。
3. 理論的研究は、現実問題に有用であることを、その価値基準とする。
4. 力のありそうな人となるべく多く議論をする。
5. 学生は、できるだけ早くかつ多く欧米に行かせ、発表・議論をさせてくる。
6. 理論を研究する学生にも、必ず実装をきちんと経験させる。
7. システム実装が得意な学生には、自分の実装の特徴を概念化・明示化する訓練をさせ、かつ実装の重要性を自覚させる。
8. 各学生には、その人生に1つでよいから、他人が考えたことのない、概念・アイデア・プロダクトを、確立あるいは作成するという使命感を持たせる。

本稿の最後に、これまでの私の学術生活の上で、大変お世話になったり、強い影響を受けたりした先生方や皆様のお名前を順不同で列挙させていただきだけで謝辞に代えさせていただく。本来なら、その由来のようなものを申し添えるべきだが、紙面の制限もありご容赦いただくことにする。

2 なぜソフトウェアの研究へ

2.1 Noam Chomsky

幾分自叙伝じみているが、私の本当の興味は「言語・言葉」にある。言語一般について、今でも強い興味をもっている（といて、何ヶ国語も読み書きで

My Software Research.

Akinori Yonezawa, 東京大学情報理工学系研究科コンピュータ科学専攻, Department of Computer Science, Graduate School of Information Science and Technology, University of Tokyo.

コンピュータソフトウェア, Vol. 21, No. 5 (2004), pp. 20-31.
[パイオニア] 2004年8月17日受付.

きるわけではない)。大学の教養学部では、第2外国語としてロシア語を選択した(1966年)。スプートニクが上がり、科学にはロシア語がいる、ドイツ語等はそのうち勉強するだろうという、安易な選択であった(我々の頃は、大学院の入試にも第2外国語があり、それをロシア語で受験せざるを得ず、まったく往生した)。しかし、我々のクラスのロシア語の先生の一人は若くて、学生に刺激を与えようとした。その先生が、あるとき1年生に向かって、ロシア語の文法を「生成文法」で記述するという話があるということ、講義の中でポロっと話した。その講義の直後、私は教師の後を追って彼の部屋に行き、「生成文法」って何ですかと訊ねた。そこで、その先生は、2つほど日本語の論文のコピーを下さり、「興味があるなら、Noam Chomsky の Syntactic Structures という本を読むといいですよ」と恥ずかしそうに言ってくれた。

さっそく丸善に行ってみたら、1冊残っていたので購入した。MITでのChomskyの学位論文である。吸い込まれるように読み進んでいくことができた。おそらく革新的な内容が存在したが、内容それ自体にはあまり感銘を受けなかった(むしろその後大学院で読んだ、言語のタイプ0、タイプ1、タイプ2...等の階層が議論されている、Information and Controlに発表された論文は強い印象を受けた)。強く打たれたのは、彼の明晰な言語表現であった。それは、ある意味で丸山眞男の「現代政治の思想と行動」を読んだときの明晰な日本語への驚きと同じであった。論理実証主義の雄Wittgensteinが「およそ語れるものは明晰に語れる。語れないものについては沈黙を守らねばならない」と言い放った気分が小気味よく思い出された。

2.2 アセンブラの解読

教養学部の学生には、一般の教養学部の講義よりレベルの高い特別な講義群がボランティアの先生方によって用意されていて、私は1.5年の間に折原明先生の「Banach空間論」、齋藤正彦先生の「超関数論」と森口繁一先生の「電子計算機入門」をとることができた。前者の2つも大変面白く勉強になったが、この種のことを研究する職業はちょっと苦しいなと思った。一方森口先生の講義は、十進の磁気コアメモリ

が64K桁ほど用意されていたOkitacという計算機の機械語とアセンブラを教えてくれるというものであった。

本郷へ進学する学科を選択する時期となり、チョイスが非常に沢山あり迷った。脳のことや人間の考えるメカニズムについては、難しくてなかなか解明できないであろうから研究するのに相当な根気がいると思ったが、自分自身の興味をそらすことは出来ず、そうしたことを比較的深く研究されていた本郷の南雲仁一先生を友人と二人でお訪ねした。「そういうことをここで勉強したいなら、エレクトロニクスと数学をよく勉強しておきなさい」という助言をいただいたが、数学はまだしも、私はエレクトロニクスは苦手であったので、この系統はあっさり諦めた。

結局、計数工学数理コースに進学し、夏学期が終わるか終わらないかで全学ストライキに突入した(1968年)。夏学期の甘利俊一先生、伊理正夫先生、森口繁一先生の講義は、私がよいと思う講義のモデルとなった。学生ストライキによる混乱と混沌の中(本郷の総合図書館が学生によって閉鎖されるまで、そこでChomskyの著作を何冊か読んだが、Syntactic Structuresを超えるものはなかった)での3,4年生であったが、その森口先生の講義の演習で使われたAlgolのサブセットAlgolipが吐き出すアセンブリ言語がどのようにOkitacの機械語になるのかに興味を持ち、Okitacのオペレータの女性に頼んでアセンブリプログラムの部分を16進でdumpしてもらい、それを一生懸命に解読した。名前表や、新しい名前が出てきたときの処理、同じ名前の出現をポインタでリンクしておくなどの処理を知って、感心した。次にAlgolipのプログラムをアセンブリに変換する部分に挑戦しかかったが、16進でまた解読する気力もなく、逆アセンブラを自分で作ってから解読するという知恵もなかった。しかし、このアセンブラの解読は、言語処理系に強い興味を持つきっかけとなった。

また、伊理先生が、演習のような講義で、「Fortranでスタックをシミュレートしながら、ハノイの塔の解を再帰プログラムで書いて実行せよ」という問題を出され、それを解いたとき、気分が「すっと」したのを今でも記憶している。

2.3 ALGOL N のコンパイラ作成

大学紛争も終わり、大学院の入試のロシア語はなんとかごまかし、森口繁一先生・和田英一先生の研究室に所属させていただいた。武市正人さんと同期・同門である。このころ、プログラミング言語研究の世界的な流れの一つに、拡張可能言語 (Extensible Languages) の設計・意味論・実装などがあり、Algol 系言語の拡張形式について数ヶ国が提案をし、その中の一つを IFIP による世界規格にするという競争が始まっていた。日本からは、故島内剛一先生を中心に、故米田信夫先生・和田英一先生が ALGOL N という言語の設計を行っていた。同様の目的で、当時日立中央研究所の中田育男先生を中心とし、別な言語名と方式での設計・実装も始まっていた。当時修士 2 年の鈴木則久さん・藤崎哲之助さんが中心となり ALGOL N のコンパイラの実装が始められ、修士 1 年の武市さんと私がその実装の手伝いをするようになった。これは、お手伝いであったが、コンパイラをどうやって作るかを知る上で、大いに勉強になった。ガーベジコレクションのことを学ぶ機会ともなった。ちなみに、拡張可能言語とは、一つの言語内でプログラマがいろいろなオペレータを定義することができ、そのオペレータ群を使えば Fortran でも Cobol でも PL/I でも、同じコンパイラでコードが生成できるようにすることを目指す言語であった。コンパイラ作成に使った言語は Fortran とアセンブリ語であった。

2.4 定理証明

先輩の修士論文が終わり、次は自分の番であった。コンパイラをもう一度作っても仕方ないので、修士 1 年のときから少しずつ勉強していた定理の自動証明のことを考えてみることにした。J. A. Robinson の導出原理やその変形・変種などたくさんの論文を読んだが、ぴんと来るものはなかった。島内先生にご相談したら、G. Gentzen の 4 つの論理系と、そのなかの特に LK に基づく証明手続きなどについて、わざわざ 2, 3 回講義をして下さった。「後はなにか、これを使っておもしろいことを考えたら」と言われて一瞬途方に暮れたが、ひと夏 on-and-off で考えていたら、修士論文の種が幸い見つかった。

一階述語論理の恒真性の判定は、一般には決定不能 (あるいは半決定可能) であるが、証明すべき論理式の形を制限すると決定可能になる場合がある。Ackermann 関数で有名な W. Ackermann の著による比較的厚い「Solvable Cases of the Decision Problem」という本のタイトルが目に入り、決定可能なケース (decidable cases) を調べてみる気になった。何週間かして、いくつかの決定可能なケースが証明できた。そのうちの 2 つは、幸か不幸か、1930 年代にすでに Jacques Herbrand が発見していたことが分かり落胆した。しかし、証明法も違出し、Herbrand が証明した定理の再発見なら、修士論文ぐらいいは構わないだろうという評価にしてもらった。理論の追求はここまでにして、M2 の 9 月に入ってから、東京女子大にあった Tosbac3400 をお借りして実装を開始した。発見した決定可能なケースを前もってチェックする機構を内蔵した、シーケント形式で一階述語論理式を書けばその恒真性を (準) 自動的に判定する定理自動証明器を一人で実装した。まともな Lisp の言語処理系はまだ日本に無かったので、再びアセンブリ語でプログラミングした。約 6000 語ぐらいのプログラムになったように記憶している。I/O は Fortran で書きリンクした。この研究は、私の最初の論文となって「情報処理」に掲載された。

この実装に関連して、木 (tree) の中のポインタを反転させることで、スタックを用いずに木をトラバースする方法を考えついた。島内先生が「リカーションとスタックとは同相なんだ」とつぶやかれたのを小耳に挟み、何か旨い方法がありそうだと思っていたので、この方法を思いついたわけである。われながらこれは良いアイデアと思って、助手の人にレポートの形でお見せしたら、「これなら出来るね」とは言って下さったが、論文にしたらとまでは勧められなかった。これは、私がもう少し積極的であればよかったと悔やまれる経験であった。なぜなら、これと全く同一のポインタ反転アルゴリズムが、Schorr-Waite-Wegbreit のアルゴリズムとして米国の著名な教科書に掲載されていたからである。このようなことは、昔の日本人の研究にはたくさんあったのではないだろうか。

3 外国研究者の訪日

3.1 John McCarthy

私が修士1年のときに、StanfordのJohn McCarthyが大学を訪問し、彼のAI研究所の当時最新の成果をフィルムで見せていった。

Instant Insanityという名称のカメラ系に連動したマニピュレータシステムシステムのデモフィルムを講演中に見せてくれた。それは、6色塗りの立方体6個を縦に積み上げ、かつ全ての側面を同色にするという知能(?)を持ったロボットシステムのようなもので、McCarthyは、これをDARPAの連中に見せて研究資金を貰うんだと言っていた。当時、同じ講演で、Lisp 1.5のmeta-circularな操作的意味論の話もしたように記憶している。私は、meta-circular definitionを動的に変化させたらプログラムが進化しそうだと考えた。これが、後年自分が自己反映計算の研究に入っていったきっかけとなったことは間違いない。このとき彼を羽田まで送りに行った際に、自分の考えている定理証明器の話をしたら、「何に使うのか? 目的がはっきりしていないならあまり意味がないよ」「導出原理に基づくものか? もしそうなら、長い証明/探索には、そのままではうまくいかないから、巧妙な証明制御が必要だ」とMcCarthyは言ってくれた。導出原理に基づく定理証明の場合、証明したい述語の記述されている形を全てconjunctive normal formに書き換えてから「導出」を始めるので、元の述語の記述構造が失われてしまうから、証明に問題領域の性質や構造が利用しにくくなるというのが、彼の見解であると私はその時推測した。

3.2 Rod Burstall

これも修士1年のころと記憶しているが、JIPDECが招いた多くのAI関係の欧米人のなかに、一人だけかなり若いEdinburgh大学のRod Burstallがいた。彼は、Donald Michieの研究所でやっている巨大で不恰好なロボットハンドのスライドをみせながら、この制御はPOP2という言語で記述され、その言語はパターンマッチングやある種のバックトラック機能を持っていると言っていたように記憶している。この

言語は、ある意味で、Carl HewittのPlanner言語の前身に近いものであった。さらに、Burstall自身の研究として、様相論理を用いてロボットの制御を記述する構想を述べていった。私が講演直後に様相論理のことをいろいろ訊いたためか、「あなたはCresswellの“Modal Logic”という教科書を読むべきだ。私もその本で勉強した。」という内容の手紙が2週間後に英国から届いた。このときも早速丸善に行ったが、この本は見つからず、注文して2ヶ月後に届いた。本にはS1だのS5だのの様相論理のシステムが意味論と共に書いてあり、Saul Kripkeが15歳で作ったという意味構造(semantic structures)には感銘をうけた(BostonでKripkeの講演を聴くチャンスに恵まれたが、そのときは、彼の昔の天才ぶりはあまり余り感じられなかった)。

3.3 Andrei Ershov

これは私が修士2年生のときだと思うが、後藤英一先生が、ロシアの理論家Andrei Ershovを招待し、東大で彼の講演が行われた。いわゆるプログラムスキーマの話で、「フローチャート風の言語で書かれたプログラムの等価性をどのように論じるか」というもので、それ自身は決して悪くない講演であった。しかし、「再帰呼び出しのあるようなプログラムの等価性は、この方式でどうするのか」と手を上げて質問すると、Ershovはかなり長い返答をしたが、正面からの回答は出てこなかった。そのあと司会者が更なる質問を促して後藤英一先生を指名されたが、後藤先生は「いまの学生さんに先に質問をされてしまった」と頭をかきながら言われたので、なんだか嬉しくなったのをやはり記憶している。

3.4 Patrick Winston

やはり修士2年のときに、当時の電子技術総合研究所(今の産業技術総合研究所)がMIT人工知能研究所のPatrick Winstonを招き、MITでの最新の研究について、比較的详细に日本で知ることのできる機会を作ってくれた。3日間、朝から夕方までWinstonはずっと講義をした。そこでは、彼の概念学習プログラムの詳細、Terry Winogradの自然言語理解システム

SHRDLU のかなりの詳細、それに、このシステムを大成功に導いた、Hewitt が考案し Gerald Sussman らが Lisp で実装した micro-Planner という言語が手際よく紹介された。Winograd のシステムは当時まさに画期的であり、AI(人工知能)の更なる大きな可能性を世に訴えるうってつけの強力さとアピール度をもっていた。一方、micro-Planner は、プログラミング言語として大変興味深いもので、バックトラック機能と、表明データの宣言、削除、パターンマッチング、パターン主導型手続き起動などの新しい機能を備えた、これまた画期的な言語であった。そして、述語論理式の「手続き的な解釈」を与える言語としても認識されていた。micro-Planner については、東大の研究室での和田英一先生のゼミで、この言語のマニュアルを読み尽くした。

4 MIT の AI Lab と LCS で

4.1 留学スカラーシップ

人工知能と micro-Planner への強い興味で、MIT に長期間行ってみたいと思い、森口・和田両先生にご相談したところ、是非行ってみるといいと言われた。その前に、和田先生からお借りした、Multics のマニュアルや Jerry Saltzer の学位論文を読んでいたのも大きな理由かもしれない。日本学術振興会は当時、米国に9ヶ月留学するためのスカラーシップを毎年募集していたので、これに応募した。受験に際し、以前そのような面接試験官をしたことがあるという方をご存知の友人のお母さまに、一般にどんなことが質問されるか伺ってもらった。また、TOEFL と GRE を受けて書類を整え、同時に米国の大学院博士課程6つに出願書類を送った。数人の先生方に推薦状をお願いしたが、どの先生も私の実力以上のまさに身に余る良い推薦状を出して下さったようである。5つの課程からアクセプトがきたので、MIT に行くことにした。アクセプトがきたのが4月のはじめで、7月には学術振興会がフルブライト委員会に委託している6週間の語学研修のために Texas 州 Austin に行かされた。それを終えると、New Orleans を回ってそのまま東海岸の Cambridge に向かった。

4.2 セミナーと講義

1973年の9月に始まった大学院の講義は、秋学期、春学期ごとに週2,3回の講義とゼミがあり、新入生には学生個別に、どの講義を取るべきかの相談と助言をするカウンセラがついた。Winston が私のカウンセラで、彼はどの講義をとってもいいと言ったが、今度新設された Albert Meyer の“Algorithms”という講義は、是非取るようにと強く勧めた。実際、Meyer の講義は、その前夜はいつも徹夜だったと彼が後に述懐していたように、彼にとっても初めての、大変“力”の入ったものであった。本質だけを非常に分かりやすく説明してくれ、かつ毎回宿題が出た。それをTAが採点して翌週には返してくれた。

セミナーは、多くの場合若手の先生が自分のリサーチトピックや得意分野を中心に、学生の人数をせいぜい7,8名ぐらいに限って出席させて、講義や学生の発表と徹底した議論を行うもので、私は、はじめの年の9月から Carl Hewitt のセミナーに毎回出席することとなった。Brian Smith, Henry Baker, Ken Kahn などの学生と Hewitt が喧々諤々、私が議論に口を挟む余地などなかったが、時々私が何かを言いかけて口をパクパクしていると、Hewitt が「Aki が何か言いそうだ」と言って皆の発言を制止してくれた。議論はもちろん Actor の揺籃期の混沌たるものであり、Hewitt が新しい操作的意味論を提案するたびに、セミナー参加者のブーイングをうけた。しかし、彼の考えはいつも斬新で直感と洞察に満ちたものであり、学生の教師評価では most brilliant professor という評価を受けていた(もちろん、1973年の時点で micro-Planner や Planner の研究は完全に終わっていた)。

Barbara Liskov の CLU 言語設計のセミナーにも参加していた。このセミナーはより整然としたものであったが、Liskov の冷静な判断と説得力で、CLU を着実に設計していき、実装も学生を雇いながら上手にやっていくのをつぶさに見た。Alan Snyder と Liskov とによる、抽象データ型の定義モジュールや、基本手続きのパラメタに型の制限をいれる where-clause の宣言導入に関する議論は、幾度となく激しいものがあった。

Lisp 1.5 のマニュアルの著者の一人である Michael Levin による「Logic for Computer Science」という講義では、講義の時間が余ったので、私が東京で研究した「述語論理の証明手続き」について、3 回連続で 20 人ぐらいの大学院生の前で発表させてくれた。Gentzen による LK に基づくシーケントベースの形式なので、米国ではあまり知られていないようで、興味をもって皆が聞いてくれたのは嬉しかった。

4.3 RA をもらう

私の日本からのスカラシップは 1 年間だけであった。その先ここに留まるためには、来年の RA (Research Assistantship) をなんとか手に入れなければならなかった。セミナーでの学生の点数は、term paper (日本のレポートよりも少しプロジェクト性が高いもの) とセミナーでの発言等で決められる。私の場合、発言はあまり勘定にいれてもらえなそうであったので、Hewitt のセミナーの term paper に焦点をあてて、彼を impress するしか方法はなかった。そのとき、どんな term paper を書いたか記憶にないが、かなり大部のものだったことを記憶している。クリスマスごろ、来年度 RA が欲しいと Hewitt に頼むと、1 月の終わりに返事をすると言われた。MIT の EE&CS 学科では Ph.D. candidate になるための資格試験は 3 回合格する必要がある。1974 年の 1 月の初めに第 1 回目 (PWE: Preliminary Written Examination) を受け、2, 3 科目においてクラスで 1, 2 番の成績であったらしい。それとどう関係があるかわからないが、1 月の終わりには、Hewitt から、来年度から君に RA をやるというメールを受け取った。その直後、いつまでもらえるのかと訊くと、「Ph.D. が取れるまで」とウインクされた。ARPA からの研究資金を AI Lab. と LCS (Laboratory for Computer Science) がちょうど獲得した時期で、私は両研究所から 50/50 の共同サポートを受けることになった。このため、所属もこの両研究所ということになった。

4.4 学位のための研究

2 年目からは RA になったので、なにか特別なことをやらされるのかと思って、Hewitt に何をすべきか

と尋ねると、「Actor 形式の中でなら好きなことを研究してればいいよ」という返答が戻ってきた。そこで、前年度 Hewitt が出していた Actor 形式の操作的意味論、すなわち meta-circular interpreter を 2, 3 週間眺めていて、これを使って Actor 形式 (まだ並列性を導入していなかったもの) で書かれたプログラムをデータなしで抽象的に評価する方式を考えついた。まずはキューに相当する Actor プログラムを、具体的なデータなしで評価するというところを行った。これを Hewitt は meta-evaluation と呼びたが、実は後のいわゆる symbolic evaluation と呼ばれるものと、同一のものであった。この話は AI Memo として公表することができ、これを読んでいた古いフランスの研究者からは、symbolic evaluation のパイオニアの仕事と言われることが何度かあった。もう少し別の見方をすると、状態を持つ (stateful あるいは side-effect を持つ) 抽象データ型 (ADT) の実装が仕様を満たすか否かのチェックをしていることに対応していた。状態を保持する抽象データ型に並列性を導入すると、後に私が「並列オブジェクト指向」と呼んだ計算モデルに自然と発展することになる。

Hewitt はこの研究を比較的気に入ってくれたらしく、この方向で学位論文を書くことを勧めだしてくれた。まだ Actor 形式自身もはっきりできていない上に、並列性を扱うモデルも曖昧であったので、道は遠いとは思ったが、この方向で進めていけばなんとかなりそうだという気もしてきた。この間の Hewitt との議論は e-mail を使って行うことが多かった。彼は何しろ夜中の 10 時ごろオフィスに出てきて朝帰宅することが多かったので、彼に会って話をするタイミングを見つけることになり苦労した。

学位のための研究は、並列性を明示的にいれた Actor 計算モデルをまず明確に定義することであった。これを表示的意味論として書き下す研究も考えたが、良い表示的意味領域を考案するような数学的な力もないので試みなかった。学位が終了したあと D. Scott からメールが来て、意味領域を考えたいから君の学位論文のコピーを送ってこないかと言われたので喜んで送ったが、その後彼自身で Actor の意味論のための領域を考察した様子ではなかった。私の後、MIT の

Ph.D. の学生 Will Clinger がこの分野で論文を書いたが、数学的な欠陥があったようである。

そこで、むしろ私は、かなり精密に定義された仕様言語を考案し、並行性をもつ Actor プログラムがその言語で書かれた仕様を満たすか否かをチェックする、symbolic evaluation に基づくかなり形式的な方法を考案した。この仕事が学位論文の主要部分になると思ったが、指導教授である Hewitt は、形式的な部分はいいが何か分かりやすい appealing な例がないとだめだと言った。彼を満足させる例を考え付くのに3ヶ月ぐらいかかったように記憶している。

4.5 よい例題 (モバイルオブジェクト・モバイルエージェント)

そのころ、ノルウェーのオスロ計算センターにいた Kristen Nygaard(2001年 Turing 賞受賞)が、最初のオブジェクト指向言語である Simula67 の後継言語を設計していて、我々と多少の交流があった。そこで、Hewitt の要求する良い例として、Actor 形式で何らかの社会システムあるいは離散事象のシミュレーションを書き下し、それについての正しさを形式的に検証するのはどうかと考えた。離散事象の例といっても、ガソリンスタンドへの車の出入りのようなものは退屈なので、もっと良いものはないかと考えめぐっていた。ある日、アエログラムを買うのに Cambridge 市の中央郵便局に行く機会があった。これを機に、少し大きめの郵便局での訪問者・局員・カウンター・ドア・局内ポスト・切手・お金などを、actor で表現することを思いついた。もちろん大部分の個々の actor は並列に挙動するわけで、並列性は十分あり、かつ分かりやすい例となった。図示すると図1のようなものだが、これを見て指導教授は OK を出してくれた。学位論文全体を書き上げるには、その後夏をはさんで半年近くかかった。学位の審査委員は3人で、指導教授の Hewitt と、Thesis Reader である B. Liskov と J.C.R. Licklider の両教授であった。Liskov は細かく提出論文を読んでくれて、コメントも沢山くれ、かなり英語も直してくれた。彼女にも大変感謝している。

この郵便局モデルは、後に「モバイルオブジェクト」や「モバイルエージェント」と呼ばれるものを

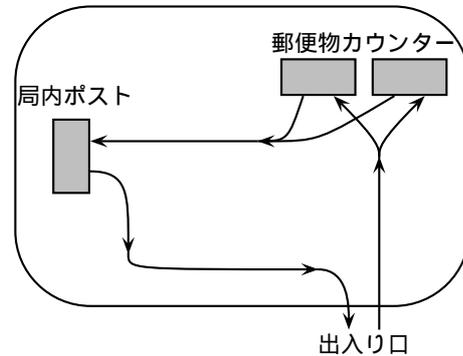


図1 Cambridge Post Office

考える上で、大変役に立った。私としては、これは、この種の話の先駆的な仕事とと思っている。局内を移動する訪問者も actor であり、それらは物理的に移動しながらメッセージを授受できるようにモデル化した。また、数年前、Boston であった理論オブジェクト指向ワークショップの招待講演でこの話をしたとき、Ambient Calculus を考案した Luca Cardelli は、Ambient と例えばカウンター actor の類似性を指摘して、Ambient のような階層・入れ子構造をカウンターは持てるのかと質問してきた。答えはもちろん yes である。上の郵便局モデルで、郵便局に出入りする訪問者をモデル化する actor は多数郵便局に存在し、かつ待ち行列をつくるなどしてカウンターからのサービスをうける。また、訪問者の状態 (state) によって挙動が変わってくる。1990年代の後半に我々の研究室で、関口龍郎氏が JavaGo を、橋本政朋氏が MobileML を設計・実装したのは、このときの郵便局モデルに対する私の思い入れを投影したものである。

5 日本に戻って

5.1 「算法表現論」

米国留学の最初1年間は、日本学術振興会からお金をもらっていたので、ピザのタイプで、2年間は日本に帰ることが決められていた。和田先生のお世話で東京工業大学の木村泉先生の研究室の助手として就職させていただいた(1978年1月)。日本の大学の計算機環境は、カードによるバッチ計算方式で、TSS は実に 300bit/sec のモデムでつなげて使用するもので

あった。メールなど及びもつかない状況であった。

ここで助手を5年近くしていたが、その間英文の論文を5つほどしか書けなかった。たとえば、関数型言語のちょっとした話や、簡単なデータベースへの問い合わせプログラムの自動合成法の話を書いた論文のほか、学位論文の内容の一部を「A Formal Specification Technique for Abstract Data Types with Parallelism」という標題でまとめた論文などである。助手であったので研究室の学生さんと一緒に仕事ができず、大変苦しい時代であった。しかし、岩波講座情報科学の12巻「算法表現論」の第2章から終章の第12章までを書かせていただいたので、精神的には平衡が保てた。また、体系的に新しいものを比較的易しく書くという点で、自分自身の勉強にも大いになった。この間、Oxfordに移ったC.A.R. Hoareに手紙を書いて、ポストクのポジションはないかと訊くと、あるから来てよといわれたが、手続きの最後の段階でうまくいかなかった。

述懐すれば、この時点で日本に留まって良かったような気がする。この時代に京大数理解析研究所の中島玲二さんや、当時電総研にいた古川康一さんとソフトウェアの学会を日本でつくる計画を練ったりもしていた。その後、多くの方々のご賛同とご努力の結果、「日本ソフトウェア科学会」が設立された。

5.2 自分の研究室を持って

東工大で助手から助教授にしてもらい、自分の研究室を作ってもよいということになった。35歳のときである。始めにしたことは、研究室助手になってくれる人を探すことであった。大変幸運なことに、京大数理解析研究所の中島さんのところで修士を取った柴山悦哉という学生がいるということで、京都まで会いに行き、その場で彼に来てもらうことにした。数理解析研究所の難しい数学の入試をこなしているし、Prolog-KABAの主たる実装者の一人で、これに関し本も出版する予定であった。最初から大変よい片腕に恵まれたのは、実に幸運であった。

大学の教師になると講義という当然の義務が生じ、おそらくどなたでも同じであろうが、はじめの1年は、担当の講義の準備に忙殺されて、研究などは何も

出来ないのが実情である。そんな中、学部4年生の大沢一郎君1人だけが私の研究室に志願してくれ、学生として配属されてきた。私は助手のときから彼の演習のレポートなどをよく見ていたが、どのレポートもきれいな字で明晰に記述され、内容的にも100点以外につけようのないものであった。自分は講義の準備のためなどで結構忙しいので、彼にTerry Winogradの学位論文と自然言語処理の適当な文献を渡し、時々議論をする程度であった。確か、私から彼には、自然言語で新幹線の予約を対話的にできるシステムが出来るところまで作ってみたら、とまで言っただけであったと思う。彼は、卒論終了の段階でかなりよく動くシステムを構築し、さらに修士課程に入り、当時として画期的な自然言語対話システムを構築してくれた。このシステムのお陰で、NHK(7時のニュースを含む)に何度か出してもらったり、日経産業新聞や科学朝日などに記事を書いてもらった。

5.3 並列オブジェクトの研究(ABCL/1の研究)をはじめ

1983年に入って、日本に戻って初めて自分の研究グループが作れるようになり、本格的に研究のできる環境が整ったとき、なにを本格的に研究するか熟考した。私が学位論文を終えるころ、HewittはActorの理論形式からなんとか状態概念を消し去ろうとしていた。もっと関数的(functional)なものにしたがって、いろいろ工夫をしていた。しかし、私はむしろ、様々な並列・並行現象のモデル化には状態がある方がより適切だと反論していた。そこで、学位論文の延長にあり、かつHewittの進みつつある路線とは異なる自分の信条に沿った、状態を持つことを基本とする並列・並行モジュール群についてじっくり研究していこうと決意した。このモジュールは後に「並列オブジェクト」と呼ばれるものである。

5.4 言語研究の三位一体あるいは正四面体と再掲論文

この並列オブジェクトが有効であることを示すためには、(1)「並列オブジェクト」群に基づいた計算・情報処理のモデルの確立、(2)そのモデルに出来るだ

け忠実なプログラムを記述するプログラミング言語の設計, (3) その言語の厳密な意味論の確立, (4) その言語の効率のよい実装 (コンパイラ等) の実現, (5) その言語の実用, を行わなければならないと私は思っていた。また, 言語を使用しやすくするための諸ツールの工夫も必要である。これら 5 つのことを, これから一つ一つ実現していこうと思ったわけである。また, これらのアスペクトを考えていく中で新しい着想に遭遇し, さらなる様々な発展が期待できると, 楽観していた。

実際, 東工大の渡部卓雄氏と一緒に考えた並列オブジェクト群における自己反映計算 (reflection) などは, その顕著な例である。MOP (MetaObject Protocol) の後を追って現在注目を受けている AOP (Aspect Oriented Programming) というモジュール分割は, 並列オブジェクト群の場合どのような興味をもっている。また, 並列オブジェクトにどのような継承機構をいれるかという議論は, Jean-Pierre Briot 氏との議論からはじまり, 松岡聡氏がこの問題を「Inheritance Anomaly」と命名した。彼が主として多様な議論を展開し, 国際的にも大きな注目を浴びた。松岡氏とこれに関して論文を書いたときは, この込み入った問題をこれ以上追求してもあまり実りの多い結果は出ない, という警鐘を鳴らすことを意図したつもりであった。しかし, 論文が刊行されてから, 「Inheritance Anomaly」を標題にした論文を数多く見るようになり, またこの問題の研究のために研究室に来たいという外国人学生が続出したのには閉口した。

2 つ上のパラグラフで言及したプログラム言語研究へのアプローチの (2), (3), (4) は, 言語研究の「三位一体 (trinity)」と勝手に命名しており, その正三角形に (5) という頂点ノードを加えて立体化したものを, これも勝手に言語研究の「正四面体」と呼んでいる。なんとなくこれらの感じを表現する単純な図をつけておいた (図 2)。

さて, 日本ソフトウェア科学会のフェロー受賞のご褒美として, 昔の論文を再掲してよいということで, 選ばせてもらった論文「オブジェクト指向に基づく並列情報処理モデル ABCM/1 とその記述言語

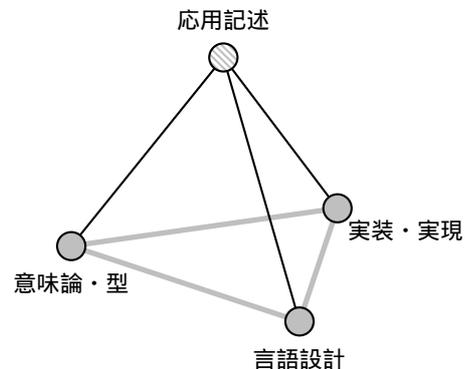


図 2 言語研究の三位一体と正四面体

ABCL/1」は, 上で述べた (1) の確立を目指した論文である。東工大での研究室の第 2 期生にあたる, この年次の最も優秀な学生さんである渡部卓雄氏, 本田康晃氏, 高田敏弘氏が, 私の研究室に配属されてきて, さらに柴山助手とフランスからの留学生 J.-P. Briot 氏が加わった大変強力なグループで日夜議論を重ねながら, 私が纏め上げたモデルと言語を論文としたものである。これは 1986 年の 4 月に書いたものであるが, ほとんど同時に内容を短くして ACM の第 1 回 OOPSLA (Object-Oriented Programming, Systems, Languages, and Applications) に投稿し, うまくアクセプトされた。これで国際舞台に復帰した気分がしたのを覚えている。もっとも, 前年にはパリとベルリンで招待講演を頼まれ, ABCL/1 の部分的な話をする機会を得ていたが, 米国での最初の第一線の舞台となると, その感慨はひとしおであった。

5.5 OOPSLA86 (Portland, Oregon)

再掲の論文にあるように, (1) の基本的な計算モデルが確立された。三位一体のアプローチとしては, (3) 意味論と (4) 言語処理系の実装・実現をなんとかやっつけてしまわなければならなかった。意味論に関しては, Hewitt の学生で自分の後輩にあたる Gul Agha が, 彼の学位論文として, 大域的名前の無限生成を許す遷移システムを基礎として, ACTOR というタイトルの成書を公刊する予定になっていた。少し焦りを感じたが, 彼の意味論は状態を持たないモジュールを基礎にしているので競争しないことにし, 人間的にも

「うま」があったので、むしろ相互訪問や論文の相互参照を繰り返して協調していくことになった。これに加えて、比較的古い年代層には名前が知られていたブラウン大学の Peter Wegner が、なんだか私を気に入ってくれたので、3人で「Concurrent Objects」の国際的なスクラムを組んで進んでいくことになった。また、日本では偶然、当時慶応大学の所真理雄氏がやはり、ある種の並行オブジェクトをアクティブに研究されており、ECOOP (European Conf. on OOP) や OOPSLA のプログラム委員会で共闘ができ、オブジェクト指向における並行性の問題は、国際的に常に大きなトピックを占めることができた。

設計した言語 ABCL/1 の実装は、始めた当時、東工大の一杉裕志氏と柴山悦哉氏が Lisp を用いて擬似並列で実行される版を製作してくれて、柴山氏が英文でうまくマニュアルを書いてくれた。このマニュアルを 30 冊ほど持って、Oregon 州 Portland で開催される第 1 回の ACM OOPSLA に向かった。私の発表は初日の午後で、Ken Kahn の講演の次にあたり、セッションチェアは Daniel Bobrow であった。発表は、自分では大変緊張したが、今までの自分の英語の講演では一番よいものであった。内容がはっきりしていたので、しゃべりやすかった。発表後、真っ先に Kristen Nygaard(この日の Keynote Speaker であった)が大仰な拍手をしながら質問者用のマイクに来て、Great work! Great work!と叫びつつ大変褒めてくれた。これは ABCL の研究が後によく参照される起因になったと今でも思っている。また、いくつか容易な質問もしてくれた。Bobrow も何か簡単な質問をしてくれた。私の講演の終わりに「ABCL/1 の言語マニュアルを 30 冊ほど持ってきたから、欲しい方にはある分だけお渡しします」と叫んでいた。私の講演がセッションの最後で、すぐに休憩に入ったが、20 人ほどの質問者が壇上の私の前に列を作って、質問やコメントをくれたり、マニュアルをもらいに来てくれたりした。これで、成功を実感した。

第 3 回の OOPSLA には渡部卓雄氏と共著の「Reflection in an Object-Oriented Concurrent Language」という論文もアクセプトされた。いまでも、この論文はかなり creative なものだと思っている。以

後 ECOOP や OOPSLA にはほとんど毎回論文がアクセプトされ、1990 年にカナダの Ottawa で開催された ECOOP と OOPSLA の合同会議では、私はプログラム委員長に選出された(プログラム委員会は、パリの Rank Xerox 社で行った)。さらに、1993 年までに、自分たちの論文だけを集めた論文集を 1 冊、Concurrent Objects の論文だけを編集した論文集を 2 冊、全て MIT Press から出版することができた。

5.6 三位一体の行方

一杉・柴山両氏の実装は比較的簡単な擬似並列でプログラム実行をさせるものであったが、本格的な並列計算機上で、大きな問題を効率よく処理する実装ではなかった。この間、私は東京大学理学部情報科学科に移り、松岡氏という新たな強力な助手を得た。彼のお蔭で、より広範な研究分野に手を染めることになり、大いに勉強にもなった。並列計算機上の本格的実装であるが、はじめ学生で後に助手になってもらった田浦健次朗氏によって、国際的に新規性の高いテクニックが考案され、それらが、AP1000 という富士通製の分散メモリ・マルチコンピュータ上で稼動するコードを生成する田浦氏のコンパイラの実現に盛り込まれた。マルチコンピュータの CPU 数は最高 512 個で、この点でも当時世界一の記録をもっていた。田浦氏が主著者でアクセプトされた米国の並列コンピュータや並列ソフトウェアの学会のうち 4, 5 個は、日本人として初めてアクセプトされたものである。

一方、八杉昌宏氏は電気工学科から情報科学の博士課程に入学してきた学生で、非常にユニークな考えをする人物であった。当時電総研にいた坂井修一氏が設計した EM4 という並列コンピュータの特徴の一つに、継続 (continuation) をハードウェアでうまくサポートする機構があった。八杉氏はこれに目をつけ、この機構をうまく用いて異なるプロセッサにある異なるプロセス/オブジェクト間のメッセージ送信やメモリ起動を大変速くする研究をしてくれた。

5.7 Linear Logic

言語研究の三位一体の最後である意味論 (+ 型) であるが、すでに述べたように、Gul Agha が状態をも

たないモジュールを仮定して、遷移システムによりその意味論を与えていた。私は、遷移システムではなく、並列オブジェクト群が状態をもち、それがメッセージを授受しながら状態を変えてゆく現象を、もっと自然に形式化したかった。当時は Robin Milner が π 計算を提唱し、英国を中心にヨーロッパでは、並列計算のモデルは π 計算がまさに帝国主義的な席卷状態であった。 π 計算を用いれば並列オブジェクトの挙動を記述できるような気がしたが、なんとなく π 計算の bandwagon に乗る気がせず、別の方法を模索していた。そこで遭遇したのが、フランスの Girard の Linear Logic であった。これは論理オペレータの種類が多いから、うまくいくかもしれないと直感した。このことを、当時助手をしてもらっていた小林直樹氏に話したら、10日もかからないうちに、Linear Logic の fragment をうまく使って、巧妙かつ自然な論理システムを作ってくれた。彼は、その後この論理システムに phase semantics を付けて、その完全性を証明した。

これは良い論文になると思い、早速ジャーナル用の論文を書き、Formal Aspects of Computing に投稿した。この間、当時まだ Imperial College にいた game semantics で知られる Samson Abramsky が我々のグループを来訪する機会があったので、この仕事をキチンと彼に説明して、納得してもらった。投稿した論文の査読者の一人に必ず Abramsky がなると予想されたからである。この甲斐あってか、投稿論文は1回の照会でアクセプトされた。驚いたことに、我々の論文がジャーナルにプリントされると、最後のページの終わりに「Recommended by Robin Milner」と記されていた。小林氏はその後、多様な型の導入とその応用に研究をシフトしていった。これで、三位一体の3つ目も達成したことになった。

5.8 応用プログラム

言語研究の正四面体を考えると、三位一体の上に、その言語を用いて問題解決を行う、あるいは応用プログラムを書くという問題が残っていた。これについては、大きな問題として、N-body 問題を Barnes-Hut の方法で計算するということと、RNA の準安定な

2次構造を全て構成するというある種の探索問題を解くということを行なった。N-body 問題では、質点の数が数万個程度のもので相当よい結果がでたが、N-body 問題専用の計算機の出現などで顕著な結果とはならなかった。RNA の2次構造の発見については、Computational Biology 誌に3編の論文を公刊することができた。

さらに、Maryland 大学のシステム研究所に招聘されて、将来宇宙での長期滞在に利用されることとなる宇宙ステーションの動力学研究(姿勢・安定制御)に取り組んだ。宇宙ステーションを構成する物理モジュールと、その物理モジュール間を結合するコネクタとを並列オブジェクトで表現し、モジュールに動的に力を加えて姿勢制御する問題をモデル化する。それによって、最終的には解析的ではなく、並列オブジェクト群によるシミュレーションで解を求めるといった問題に挑戦した。この問題は2ヶ月の招聘期間では終わらず、また招聘してくれた Krishna Prasad 教授の大学院生が急に大学をやめて就職することになり、このプロジェクトは立ち消えとなってしまった。いまでも、再開したいプロジェクトである。

6 「並列オブジェクト」の次は？

6.1 型・型推論

我々の学科では、演習(必修)でまず Scheme と ML 言語(あるいは OCaml)を教える上に、私の3年生への講義「言語モデル論」では ML の型推論や多相型の理論を、加えてコンパイラの演習では ML 風の言語のコンパイラを作らせるところまで教える。当然の結果として、型について比較的強い興味を持って研究室にやってくる学生が多い。一方、型システムの研究は近年相当に進歩している。通常の型のみならず、実行時のメモリ管理・変数スコープ・オペレーション適用順序などの分析まで型システムで表現し、それをソースコード分析に用い、メモリ使用の最適化・不正使用の防止などに使おうとする研究が行われている。有用な理論は、歓迎すべきものである。しかしながら、自分自身でそのような型理論を展開していく力をもう持ち合わせないので、この方面では、小林直樹氏や住井英二郎氏が去った今、創造的な研究の継続には

多少困難を感じているのが実情である。

6.2 ソフトウェア・セキュリティと実用研究

1999年11月、文部科学省の科学研究費補助金に、特定領域「社会基盤としてのセキュアコンピューティングの実現方式の研究」の代表者として申請し、幸い2000年9月から3年半の期間、6大学9つの研究グループの構成で、年間1.2億円の研究資金を得ることができた。この領域研究は、基本的には、ネットワークでつながったサーバーに存在する基本ソフトウェアの安全性を保証するための方法・手段を研究・開発するものである。この特定領域の代表者をさせていただき、またこのグループ研究の参加者の研究成果を勉強させていただき間に、ソフトウェア・セキュリティについて私自身比較的詳しくなってきたので、今後はソフトウェア・セキュリティの研究を発展させたいと思っている。

最後に、これまで長年多くの論文や著書を発表してきたが、この年齢になると現実の世界にもっともっとインパクトのある研究開発に携わりたいという欲求が強くなっている。もちろん論文のための論文を書いてきたつもりはないが、ある意味で論文以上の新しい概念と技法を体現した、現実の社会にインパクトの強いソフトウェアを構築する研究により大きく注力していきたいと思っている。

7 おわりに代えて

以上、私事を多少交えながら、私の研究を省みさせ

ていただいた。古い記憶にたよった部分も多く、多少の勘違いもあると思うので、それについてはご容赦をお願いしたい。

謝辞

つぎに列挙させていただきます先生方や皆様(敬称略)は、私のこれまでの研究生活や学術的な方向付けに大変貴重な影響を与えてくださいました。ここに記して、心底より感謝申し上げます。菊池誠、森口繁一、和田英一、島内剛一、甘利俊一、伊理正夫、五十嵐滋、長尾真、淵一博、片山卓也、森村良典、Carl Hewitt、Kristen Nygaard、Barbara Liskov、Albert Meyer、Gerald Sussman、Vaughan Pratt、J.C.R. Licklider、井上博允、Patrick Winston、Hilary Putnam、Mike Gordon、Gul Agha、Alan Schnieder、Henry Baker、Eliot Moss、Rod Burstall、C.A.R. Hoare、Robin Milner、Krishna Prasad、伊藤貴康、鈴木則久、大堀淳、宮内久男、Dennis Tsichritzis、Brian Smith、Gregor Kiczales、Guy Steele、Luc Steels、中島玲二、古川康一、柴山悦哉、竹内彰一、上田和紀、松岡聡、小林直樹、田浦健次朗、増原英彦、加藤和彦、渡部卓雄、関口龍郎、住井英二郎、大山恵弘(順不同)。本稿を綿密に読まれ文章上の改良点を多数ご指摘いただいた上田和紀先生に深謝する。

並列オブジェクトによる大規模システムの実現——Second Life システム, Twitter システム, 分子動力学アプリの場合——

米澤 明憲[†] 前田 俊行[†]

Large Scale Applications of Concurrent Objects ——Cases for Second Life System, Twitter System, and Nano-Bio Applications——

Akinori YONEZAWA[†] and Toshiyuki MAEDA[†]

あらまし 1970年代から米澤を中心に研究が始められた「並列オブジェクト」の概念や、それに基づいたソフトウェアシステムの構築法が、最近になりばく大な実ユーザがリアルタイムで利用する Web 系システムやスーパーコンピュータ上の実用アプリケーションの開発に使われるようになった。ここでは、そのような開発を可能とする並列オブジェクトの特徴的な機能や性質を明らかにする。また、到来し始めたマルチコア、メニコア時代のプログラミングモデルの一つとしての並列オブジェクトモデルにも言及する。

キーワード 並列オブジェクト, 並列プログラミング, Web アプリケーション, スパコンアプリ, マルチコアアーキテクチャ

1. ま え が き

我々は今、大量の MPU がラップトップから超高性能コンピュータまでを構成していく、いわゆるマルチコア・メニコア時代の入り口に立っている。そして、スパコンの飛躍的進歩とあいまって、コンピュータの処理能力も今後爆発的に増大しこれが更なる情報爆発を生む時代に突き進んでいる。

更に爆発する情報・データを制御し、その中から有用な情報の抽出・加工を可能にするのは、やはり処理能力の爆発である。処理能力の爆発はコンピュータのハードウェア性能の爆発だけでは獲得できない。いうまでもなくそれには、大規模な並列・分散処理のためのプログラミングやソフトウェア開発方式の確立が不可欠である。

本論文は、VLSI の量産の可能性が示唆され始めた今から 30 年以上前の時代に米澤が考えた「並列オブジェクト」という概念のいくつかの側面を紹介するものである。その当時から遠い将来、印刷物を刷るようにコンピュータ (CPU) が生産され、それが安価に

利用できるようになったとき、我々は一体どのようなプログラミング方式によって、そのばく大な処理資源を効率良くかつ容易に利用することができるのか。そのことを考えた結果が「並列オブジェクト」である。

2. 並列オブジェクトとは

並列オブジェクトを、直観的に理解してもらうための準備として、ソフトウェア部品の一つである「オブジェクト」の概念を簡単に説明する。

2.1 ソフトウェアシステムの構成

メールサーバ、ファイルサーバ、あるいは Web サーバなどのいわゆる「サーバ」や Windows や Linux のような OS などは、プログラムの集まりである、ソフトウェアシステムである。もちろん様々な分野で開発される応用ソフトウェアもソフトウェアシステムである。

一般に、ソフトウェアシステムは多数のモジュール、すなわちソフトウェア部品から構成される。これは、車が多数のパーツ・部品を組み合わせで作られているのと同じである。例えば、Mozilla や IE などのようないわゆる Web ブラウザはかなり大きなソフトウェアシステムである。これらは、図 1 にある方形の箱で示したような多数のモジュールから構成されており、例

[†] 東京大学大学院情報理工学系研究科コンピュータ科学専攻, 東京都 School of Information Science and Technology, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-0033 Japan

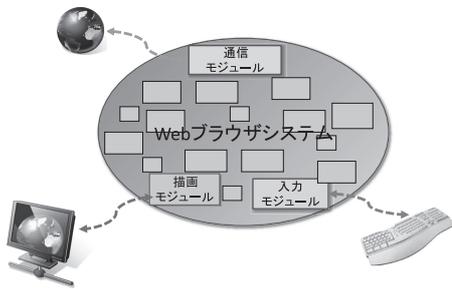


図 1 ソフトウェアシステムは多数のモジュール（部品）により構成される。

Fig.1 Software consists of many modules.

例えば通信，キーボード入力処理，ディスプレイへの出力などの機能やブラウザ特有の機能を実現する種々のモジュールがある。

2.2 モジュール（ソフトウェア部品）の要件

サイズが大きく強力な機能を備えたソフトウェアシステムを，仕様どおりに効率良く構築し，それを低コストで維持・保守するためには，システムを構成する多数のソフトウェア部品が，以下に述べるような性質を満たす，すなわち高いモジュール性をもった「形式」に従って開発されている必要がある。

- (1) 部品は，明確なインタフェースをもつ
- (2) 部品は，それが組み込まれる位置の周辺にある他の部品からの独立性が高い
- (3) 部品は，他の部品と組み合わせて，より大きい部品を作ることができる
- (4) 部品は，同等な機能をもつ他の部品によって置き換えることができる

計算機の黎明期から，あるサイズ以上のプログラムを開発するために，サブルーチン，手続き，関数などの機能がプログラミング言語に備えられた。これらの機能を実現するプログラム部分が，ソフトウェア部品，すなわちモジュールとみなされ，これらをメインプログラムから呼び出すことによりソフトウェアシステムが構成・実行される。

更に，1970年代半ばごろに設計されたプログラミング言語では，アプリケーション領域からの必要に応じてプログラマ自身が新しいデータ型を定義・実現できるようになった。これが抽象データ型である。

2.3 オブジェクト

1990年代半ば以降から開発されている大半のソフトウェアシステムは，いわゆる「オブジェクト指向」法に基づいて開発されている。この方法は「オブジェ

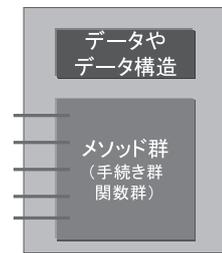


図 2 オブジェクト形式
Fig.2 Object format.

クト」と呼ばれる形式のモジュールを組み合わせるシステムを構築していくものである。オブジェクトと呼ばれるモジュールの形式は，基本的には前項で述べた抽象データ型を実現するソフトウェアモジュールの形式で，その構成はおおよそ図 2 のようなものである。

オブジェクト形式のモジュールは，ひと固まりの変数やデータ構造とそれに対して参照・更新などの操作をするメソッドと呼ばれる手続き・関数群で構成され，このモジュール内にあるデータは，ここで定義されているメソッドを用いる以外にはアクセスできない。

データは指定されたメソッド以外では操作することは許されず，また各メソッドもその呼出し（あるいは起動）の仕方が定められている。オブジェクト形式は，この意味でデータとそれを操作する手続き群が密閉あるいはカプセル化され，ソフトウェアシステムのモジュールを形成している。（ここではプログラムのコードを共有・再利用するためのオブジェクト指向言語特有のインヘリタンス（継承）機構については割愛する。）

2.4 並列オブジェクト

このオブジェクト形式を一般化し，並列・分散処理に向けた計算モデル・プログラミングモデルの基礎となったものが「並列オブジェクト」である。別の言い方をすれば，並列オブジェクトは分散・並列処理を行うソフトウェアシステムの部品形式のことである。

並列オブジェクトは，直観的に説明すると，オブジェクト形式に（仮想的な）CPU を埋め込んだ形式と見ることができる（図 3 参照）。

すなわち並列オブジェクトは，データとそれを操作する手続きと操作を実行する CPU が一つのモジュールとして密閉されたものである。一つの並列オブジェクトは，一つのオブジェクトに一つの CPU を埋め込んだものとみなせる（図 4 参照）。

並列オブジェクトでは，その中に密閉されているメソッドを起動するには，メソッドの名前を指定したメッ



図 3 並列オブジェクトの直観的説明
Fig. 3 Intuitive explanation of concurrent objects.

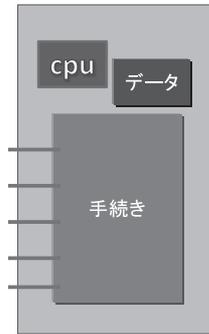


図 4 並列オブジェクト形式
Fig. 4 Concurrent object format.

メッセージをその並列オブジェクトに送る。メッセージが到着すると、並列オブジェクトは、指定されたメソッドを自分のもつ CPU を用いて実行する。

並列オブジェクト群で構成されるソフトウェアシステムでは、並列オブジェクト同士のメッセージのやり取りで、計算・情報処理が進行する。その際、メッセージ送信は基本的には非同期的で、メッセージを送った並列オブジェクトはそのメッセージがあて先である並列オブジェクトに到着し、指定したメソッドが起動されることを確認することなく、メッセージ送信以降の実行を直ちに継続する。この非同期通信によるメソッドの起動は、ソフトウェアシステム内の計算・処理の並列度を高め、モジュール（並列オブジェクト）間の独立性を強めるのに大きく貢献する。

2.5 並列オブジェクトによるシミュレーション

1970 年代半ばころ、米澤が並列オブジェクトを構想するときにはある構想が念頭にあった。将来、計算機は様々な問題領域内の世界をモデル化し、それをソフトウェアシステムとして表現・実行し、その世界を丸ごとシミュレーションするであろう。そのとき、その世界に登場する事物を並列オブジェクト群で表現し、事物間のインタラクションを並列オブジェクト間のメッセージの伝送で表現・シミュレートすれば、世界のもともとの構造をあまりひずませることなくソフトウェアシステムとして表現できるのではないか。こ

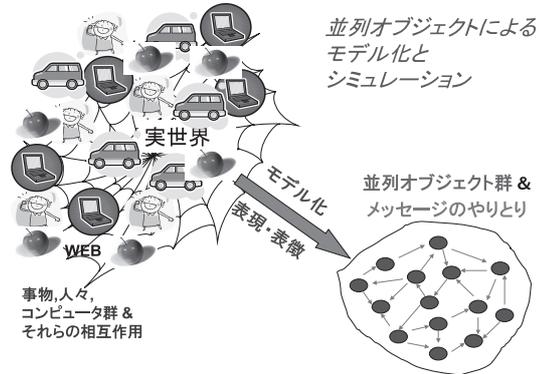


図 5 並列オブジェクトによるモデル化とシミュレーション
Fig. 5 Modeling and simulation with concurrent objects.

の考えはそれほど大きな間違いはなかったと思われる（図 5 参照）。

3. 並列オブジェクトの特性

本章では、前章でその概要を述べた並列オブジェクトについてより詳しく説明する。具体的には、六つの特徴的機構と性質について説明する。

3.1 メッセージ駆動型（message-driven）処理

並列オブジェクト内に密閉されているメソッド（手続き）群を起動するには、並列オブジェクトにメッセージを送る必要がある。メッセージを受け取った並列オブジェクトは、メッセージの中に指定された名前に従って、その名前をもつメソッドを起動し実行する。起動されたメソッドの実行が終了すると、その並列オブジェクトは次のメッセージが到着していれば、そのメッセージに従って次のメソッドの実行を始めるが、メッセージが到着していなければ、並列オブジェクトは休眠状態に入る。このように並列オブジェクトは基本的には、メッセージが到着することがきっかけで計算・情報処理が引き起こされる、メッセージ駆動型の処理方式をとる（図 6 参照）。

3.2 非同期メッセージ送信

並列オブジェクト間のメッセージ送信は基本的に非同期的（asynchronous）である。非同期通信に加えて、実行をいったん停止し、何らかのメッセージが到着すれば実行を再開するという処理が許されれば、非同期通信をもとに、同期通信も実現できることはよく知られている（図 7 参照）。また、並列オブジェクトに定義されているメソッドを起動するための非同期的メッセージ送信とともに、単純に値や情報を並列オブジェ

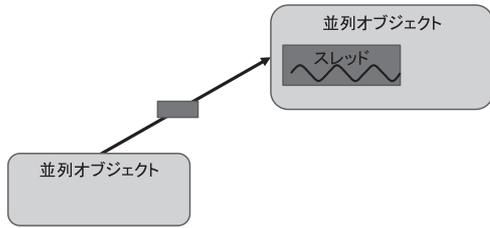


図 6 メッセージ駆動処理
Fig. 6 Message-driven processing.

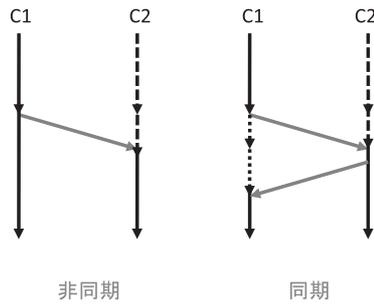


図 7 同期/非同期通信
Fig. 7 Synchronous/asynchronous communication.

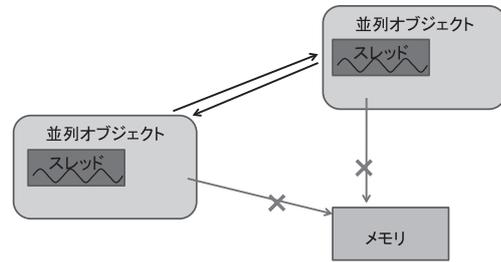


図 8 並列オブジェクト間の情報交換
Fig. 8 Information exchange between concurrent objects.

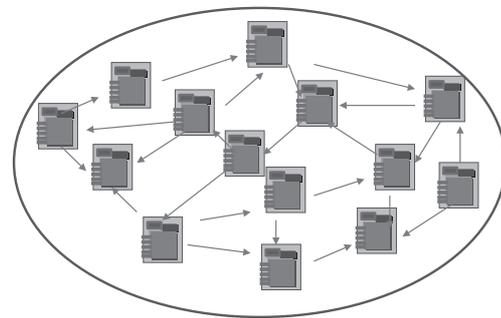


図 9 並列オブジェクトによる並列化
Fig. 9 Parallelization with concurrent objects.

クトに渡すための非同期のメッセージ送信もある。

3.3 情報の交換はメッセージ送信のみ

並列オブジェクト同士の情報交換は、転送されるメッセージを介するという手段のみが許される。この結果、プログラムシステムが並列オブジェクトのみで構成されるなら、ソフトウェアモジュール間の情報のやり取りは、並列オブジェクト間のメッセージのやり取りのみによって行われる。

つまり、図 8 にあるように、並列オブジェクト間のメッセージのやり取りによる情報交換は許されるが、並列オブジェクトのメソッド実行中に read/write 操作を直接共有のメモリに施すことは許されない(図 8 の × 印)。

3.4 多数の並列オブジェクトによる超並列性

並列オブジェクトの枠組みでは、メッセージ通信が非同期でかつメッセージの到着により並列オブジェクトが起動される。このため、並列オブジェクトが例えば二つあれば、それぞれが一つの仮想 CPU (スレッド) をもつので、この系の並列度は 2 となる。このように、並列オブジェクトをモジュールとして構成されるソフトウェアシステムでは、使われている並列オブジェクトの数だけの並列性が系の中に存在し得ることになる。これまでの説明では、並列オブジェクト内には一つ

だけの仮想的な MPU (あるいはスレッド) があることを前提としている。この前提をくずし、複数のスレッドも許すような亜種の並列オブジェクトの枠組みもあるが、その場合並列オブジェクト内の並列性を許すことになり、同一オブジェクト内でスレッドの排他制御やデッドロック回避など、伝統的な並列プログラミングに内在する問題点を引き継ぐことになり、並列オブジェクトの利点が少なからず減少することになる。

3.5 安全な軽量プロセス (lock, unlock 操作が不要)

伝統的な並列プログラミングでは、複数のスレッドが同じデータにアクセスする場合に、古いデータや更新中の中間状態のデータを読み出してしまうような、いわゆるレース状態が引き起こされたり、スレッド間の同期の不具合からデッドロックを引き起こす危険がある。これを防ぐために、プログラマは lock/unlock 操作を用いて注意深く並列プログラミングを行わなければならない。しかしながら、多数のスレッドを利用し高度な処理をする場合には、レースやデッドロックが決して生じないという確信をプログラム開発者がもつことは大変困難である。

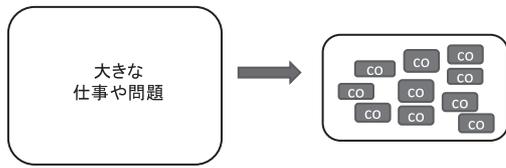


図 10 並列オブジェクトを用いた問題の分割
Fig. 10 Division of a problem by concurrent objects.

並列オブジェクトを用いた並列プログラミングでは、複数のスレッドによる共有データへの直接のアクセスは存在しない。また並列オブジェクトのメッセージ送信の方式によって、lock/unlock 操作を使うことなく同期を実現できる。このために、並列オブジェクトをベースにした並列プログラミングでは、レース状態やデッドロックが生じないプログラムを開発することが容易に可能になる。

3.6 並列オブジェクトを用いたモデル化と実装

2.5 でも触れたとおり、並列オブジェクトの考え方は、本質的に複数の事物が並列的に動作・相互作用（インタラクション）する実世界の諸問題のモデル化に適していると考えられるが、実際にモデル化したモデルをソフトウェアシステムとして効率的に実装・実行できるのは、一つの大きな問題を、相互作用が局所的なものに限定されるような大量の事物に分割できる場合である。

例えば、大量の粒子・物質の物理的シミュレーション（多体問題など）や、自律的に動作する複数のエージェントを用いることで複雑なシステムをモデル化するマルチエージェントシステムなどを、並列オブジェクトを用いて実装することが可能である。

また、問題の分割をより抽象化して工夫することで、更に多くの問題を並列オブジェクトの考え方でモデル化・実装することが可能である。例えば、CKY 構文解析や遺伝的アルゴリズムなどは、データ間の競合などを考慮してうまく問題を分割することで、並列オブジェクトを用いて効率的な並列化が実現できる。

次章では近年行われた並列オブジェクトの大規模な応用例をいくつか紹介する。

4. 大規模応用例

並列オブジェクトをベースに、米国を中心にそれぞれ特徴的な三つの大規模システムが構築されている。

4.1 Second Life システム

リンデン社の Second Life システムは、世界で数百万以上のユーザをもつ 3D 仮想世界構築ツールで、経



Image from "Programming Second Life with the Linden Scripting Language" by Jeff Heaton (<http://www.devx.com/opensource/Article/33905>)

図 11 Second Life システム
Fig. 11 Second Life system.

済活動、教育プログラム、種々の娯楽等の分野で利用され、発表当時に大きな話題となり、現在もいろいろな分野で着実に利用されている。

このシステムでは、ユーザが構築する仮想世界におけるアバターをはじめとする登場物及び登場物間の相互作用は、それぞれ並列オブジェクトとその間のメッセージのやり取りとしてプログラミングされるように、ユーザが使う記述言語（スクリプト言語）と豊富なライブラリが用意されている。

Second Life システムで、ユーザに提供されるプログラミング形式（言語）として並列オブジェクトが採用されている理由は、並列オブジェクトがもつ自然で強力なモデル化能力にある。Second Life の（仮想）世界の各登場物は状態とそれに依存する行動からなるが、それらは並列オブジェクト内のデータ・値群と、メソッド群で容易に表現できる。このため一つの単体としての登場物を、並列オブジェクトという単体のソフトウェアモジュールとしてモデル化・表現することは直接的で自然である。

4.2 Twitter システム

140 文字以内の短いメッセージ（つぶやき）を仲間やコミュニティに送り合う Twitter システムは、既に新しいコミュニケーションのメディアになりつつある。米国の大統領の意見や国際的な草の根的世論を伺い知る手段から、学会などでの講演に対する意見や反応を聴衆の間で即時に交換するツールとして活用されている。

このようなシステムには、最大で毎秒数万個を超え

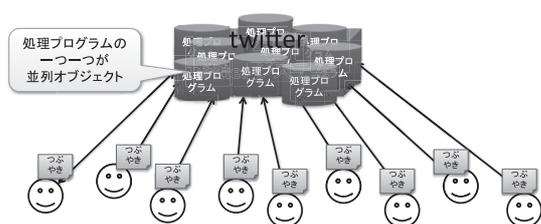


図 12 並列オブジェクトによるつぶやきの処理
Fig. 12 Processing tweets with concurrent objects.

るメッセージを即時にそれ以上の数のユーザに配信する処理を可能にする技術が要求される。

Twitter システムのバックエンド、特にメッセージキューシステムのプログラムを作成した Robey Pointer によれば [7], ユーザからの接続ごとに並列オブジェクトを生成し、その並列オブジェクトにつぶやきを処理させている。ここで重要なことは、このつぶやき処理の中核 kestrel [8] は、Scala 言語の並列オブジェクト (Actor) ライブラリを用いて、1500 行程度のプログラムで記述されている点である。このように、並列オブジェクトを用いることで、高速な処理と簡潔なプログラム記述が可能となっている。

4.3 分子動力学シミュレーションプラットフォーム (NAMD)

イリノイ大学の計算機科学者のグループと理論生物学者のグループが共同開発した、スパコン向けのナノスケール分子動力学のシミュレーションプラットフォーム NAMD は、並列オブジェクトをベースとしたプログラミング言語・実行系 Charm++ [9] によって実装されている。

Charm++ は同大の Sanjay Kale 教授を中心に 1990 年代から開発が開始され、最近では分子動力学のみならず、量子化学、天文学のシミュレーションプラットフォームの開発にも使われている。その実行時系は、並列オブジェクトの並列計算機内のノードをまたがる移動により動的分散を行わせるフレームワークも提供しており、並列オブジェクトシステムとしても先進的である。Charm++ を用いて開発されたアプリケーションは、2008 年には米国の二つの主要スパコンセンターが保有する資源の 10% から 20% を占めるに至っている。また、米澤らが 1980 年代半ばに提案しその後の様々の角度から研究・開発を行った言語系 ABCL [3] は、Charm++ に少なからぬ影響を与えている。

テキサス大学のスパコンセンターでは 2009 年の前

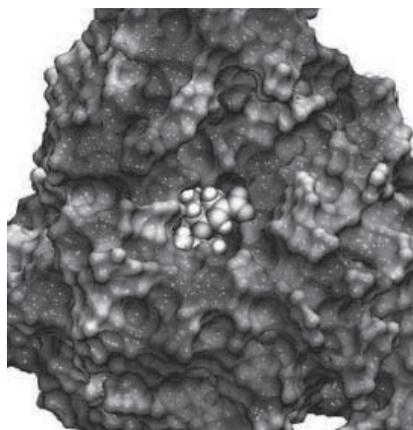


Image from <http://www.ncbi.nlm.nih.gov/rrn/RRN1015>

図 13 H5N1 鳥インフルエンザノイラミニダーゼとタミフルのシミュレーションの可視化

Fig. 13 Visualization of simulation of H5N1 avian flu neuraminidase with Tamiflu.

半に NAMD を用いて新型ウイルスの分子動力学的解析を行い、その抗タミフル性を解析し顕著な成果を得たことが報道されている。

このように、近年の米国の HPC (高性能計算) の研究分野では、大量の MPU 上で動作するプログラムを効率的に作成・実行するための新しいプログラミング言語・処理系の模索が (並列オブジェクトに限らず) 幅広く行われている [9], [11] ~ [14]。

例えば、X10 [11] は DARPA の HPCS プログラムのもとで IBM が研究開発しているプログラミング言語である。X10 は Java プログラミング言語のサブセットを、並行性や非同期通信・計算、大域アドレス空間分割などで拡張したようなプログラミング言語であり、Blue Gene を含む将来のスパコン上で実用することを目指している。また同様に Cray は Chapel というプログラミング言語の研究開発を行っている [12]。並列オブジェクトは、このような新たなプログラミング言語研究・開発の先駆的存在であり、今後その重要性は更に増していくものと予想される。

5. む す び

本論文は並列オブジェクトに関するこれまでの研究のうち、一般人にも理解しやすい部分を紹介したものである。これらの研究により米澤は、2008 年 7 月には、国際オブジェクト技術協会 (AITO, スイス・ベルン在) からダール・ニゴール賞を受賞、また 2009 年

11 月には紫綬褒章を受章した。これは、共同研究者である元スタッフや学生諸氏に負うところ極めて大である。特に、柴山悦哉, J.-P. Briot, 渡辺卓雄, 松岡聡, 小林直樹, 田浦健次朗, 増原英彦の諸氏にこの場を借りて、改めて深謝させて頂く。

文 献

- [1] 米澤明憲, “私のソフトウェア研究,” コンピュータソフトウェア, vol.21, no.5, pp.20–31, Oct. 2004.
- [2] A. Yonezawa, J.-P. Briot, and E. Shibayama, “Object-oriented concurrent programming in ABCL/1,” Proc. OOPSLA86, pp.258–268, Oct. 1986.
- [3] A. Yonezawa, ABCL: An Object-Oriented Concurrent System, MIT Press, 1990.
- [4] A. Yonezawa, Specification and Verification Techniques for Parallel Program based on Message Passing Semantics, TR191 (Ph.D. Thesis), MIT Laboratory for Computer Science, 1977.
- [5] 米澤明憲, “ACTOR 理論について,” 情報処理, vol.20, no.7, pp.580–589, 1979.
- [6] A. Yonezawa and M. Tokoro, eds., Object-Oriented Concurrent Programming, MIT Press, 1987.
- [7] R. Pointer, private communication, Aug. 14 2009.
- [8] Kestrel system. <http://github.com/robey/kestrel/tree/master>
- [9] S. Kale and S. Krishnan, “CHARM++: A portable concurrent object oriented system based on C++,” ACM SIGPLAN Notices, vol.28, no.10, pp.91–108, Oct. 1993.
- [10] M.T. Nelson, W. Humphrey, A. Gursoy, A. Dalke, L.V. Kale, R.D. Skeel, and K. Schulten, “NAMD: A parallel, object-oriented molecular dynamics program,” International Journal of High Performance Computing Applications, vol.10, no.4, pp.251–268, 1996.
- [11] P. Charles, C. Grothoff, V. Saraswat, C. Donawa, A. Kielstra, K. Ebcioğlu, C. Von Praun, and V. Sarkar, “X10: An object-oriented approach to non-uniform cluster computing,” Proc. OOPSLA2005, pp.519–538, 2005.
- [12] Chapel: The Cascade High-Productivity Language, Cray, <http://chapel.cray.com/>
- [13] Fortress, Sun Microsystems, <http://projectfortress.sun.com/>
- [14] CUDA, NVIDIA Corporation, http://www.nvidia.com/object/cuda_home.html
- [15] Second Life, Linden Ltd. Co, <http://secondlife.com/>
- [16] J. Purbrick and M. Lentzner, “Second life: The world’s biggest programming environment,” Invited talk, OOPSLA 2007, Portland Oregon, Oct. 2007.
- [17] C. Hewitt, “A universal actor formalism for artificial intelligence,” Proc. IJCAI, Stanford, pp.235–245, Sept. 1973.
- [18] C. Hewitt and H. Baker, Jr., “Laws for communicating parallel processes,” Proc. IFIP Congress, 1977.
- [19] G. Agha, A Model for Concurrent Computation in Distributed Systems, MIT Press, 1987.
- [20] 米澤明憲, “並列オブジェクト指向言語 ABCL/1 による並列処理記述とその枠組みの研究,” 信学論 (D), vol.J71-D, no.8, pp.1415–1422, Aug. 1988.
- [21] 米澤明憲, 柴山悦哉, J.-P. Briot, 本田康晃, 高田敏弘, “オブジェクト指向に基づく並列情報処理モデル ABCM/1 とその記述言語 ABCL/1,” コンピュータソフトウェア, vol.2, no.1, pp.9–23, April 1986.

(平成 22 年 2 月 18 日受付)



米澤 明憲

1970 東大・工学卒。1977 MIT 計算機科学科博士課程了。Ph.D. in Computer Science。現在東京大学情報理工学系研究科教授。日本ソフトウェア科学会理事長, フェロー, 功労賞受賞, ドイツ国立情報科学技術研究所 (GMD) 科学顧問, 内閣府総合規制改革会議委員を歴任, 現在 (独) 産業技術総合研究所情報セキュリティ研究センター副センター長, 東京大学情報基盤センター長を兼務, 21 期日本学術会議会員。マイクロソフト本社 Security Academic Advisory Board メンバ。2008 年国際オブジェクト技術協会 (AITO) Dahl-Nygaard 賞受賞, 2009 年船井業績賞, 2009 年秋紫綬褒章。



前田 俊行

2006 東京大学大学院情報理工学系研究科博士課程了。博士 (情報理工学)。現在, 東京大学大学院情報理工学系研究科コンピュータ科学専攻助教。