

離散時間共有型協調作業支援システム

木本 陽介 服部 進実
金沢工業大学 工学部 情報工学科

本論文では、離散時間共有型協調作業支援システムについて述べる。このシステムは、従来のグループウェアにおける作業共有が時間を共有することで実現されていることに注目し、メンバー全員を特定の時間に拘束せず、作業共有を作業共有時間外においても実現することを目的としたものである。これを離散時間共有と定義する。また、作業共有は発せられた意見の流れ、意見の二次利用も重要であると考えられる。そこで、ユーザから発せられた意見を作業イベントとし、イベントオブジェクトを定義することで、意見の再利用を可能とする。これを実現するために、著者らは統合作業支援プラットフォーム“Fusion”を開発した。FusionはObjectScriptによってネットワーク透過なオブジェクトマイグレーションを実現している。本システムは、Fusionを利用して意見の流れを木構造によって表現を行なう。木構造によって、任意に協調作業を再現可能にし、離散時間における作業共有を実現する。今回は、例として、以上のシステムを利用して試作した離散時間共有会議アプリケーションを用いてその評価を行なう。

1 はじめに

近年、グループウェアが注目され、オフィス業務における協調作業への利用が期待されている。グループウェアはメール回覧、スケジュール管理、データ共有、協調作業等の支援を目的としている。しかし、その利用の中心は情報の回覧であり、協調作業への利用は十分に行なわれていない。これは、協調作業そのものをコンピュータ上で実現することと、一般に行なわれてきたグループワークとの間に大きな差が存在しないためと考えられる。例えば、コンピュータを介してのリアルタイムな会議は、普通に電話を利用している場合と大きな差は存在しない。これを協調作業環境における会議システムへ対応させると、ネットワーク上での会議を実現することに成功したが、会議の生産性を向上させるには至っていない。これは、会議終了後、内容をまとめ、文書化または具体化する作業を行なう必要性と作業時間を共有する必要性が¹存在するためである。また一方で、協調作業環境で利用されるデータ形式は、テキスト、イメージ、

音声、動画などコンピュータの性能向上とともに多様化してきている。しかし、これらのデータは、単体の利用に留まっており、協調作業環境を介して得られたデータは、後に専用のアプリケーションを利用し個人で作業を進めることになる。

本論文で述べる協調作業支援システムは、著者らが開発した統合作業支援プラットフォーム Fusion上に構成される。Fusionは、ObjectScriptによって分散オブジェクト環境を実現しており、システムで利用されるデータを他の作業への再利用環境を実現している。また、データはユーザへの作業支援機能を包含し、拡張する必要がある。著者らはこれをメディアオブジェクト（以下 MOB と呼ぶ）と定義している。また、ユーザが行なう作業は全てイベントオブジェクトとして管理され、その集合を利用することで会議の再現、及び作業の再利用が可能になる。

本論文では始めに、協調作業における時間の共有について述べる。次にプラットフォームとなる Fusion について述べ、今回試作を行なった会議アプリケーションの構成を述べる。最後に、実装状況について述べ、イベントオブジェクトを利用した離散時間共有環境の評価を行なう。

Distributed Time Shared Group Work Support System
Yousuke KIMOTO Shimmi HATTORI
Department of Information and Computer Engineering
Kanazawa Institute of Technology

2 作業空間における共有性

作業空間における共有性は、時間の共有と空間の共有の2つの共有性を考慮する必要がある。前者は作業共有時の各作業者の時間的拘束を、後者は作業空間共有を意味する。両者とも協調作業空間を実現する上で当然の機能だが、重要な要素である。本章では、時間と作業の共有関係について述べる。

2.1 時間の共有

協調作業は対面型協調作業と非対面型協調作業の2つに分類される。対面型協調作業は一般にオフィス内業務として行なわれている。この対面型協調作業においては作業者が一同に介しているため、作業中の時間は作業者間において共有されている。これに対し、ネットワークを介した分散環境に見られる協調作業が非対面型協調作業である。非対面型協調作業においては、作業空間が分散されているが、作業者は個々の作業空間において時間の共有を行なう必要がある。つまり、協調作業は、対面型協調作業、非対面型協調作業を問わず、時間の共有を必要とする。これは分散環境における作業が、コンピュータを利用して行なわれ作業空間の分散化は実現されているが、時間の共有を必要とすることから対面型協調作業との大きな差が存在しないことを意味している。

この時間の共有を必要とする協調作業において幾つかの問題が発生する。

- (1) 協調作業参加者が途中で退席
- (2) 協調作業参加者が遅れて出席
- (3) 協調作業参加者が部分的作業にのみ出席
- (4) 協調作業参加者が欠席

これらの場合に発する問題点は、(1)は作業がどのような過程を経て結論に至ったかが理解できない。(2)導入部が何なのか理解できず、作業の流れを把握することが困難である。(3)は導入部が理解できず作業の結果がどのようなもの

であったかを理解できない。(4)は作業終了後にまとめられた報告書より全体を把握することは可能だが、(1)(2)(3)同様に、作業全体の流れを把握することが困難である。いずれの場合も協調作業の始めから最後まで参加しなかったために、作業がどのように流れたかを理解することが困難であることが共通している。よって、分散環境においては、作業空間の分散化だけでなく、時間の分散化を行なう必要があると考えられる(図1)。

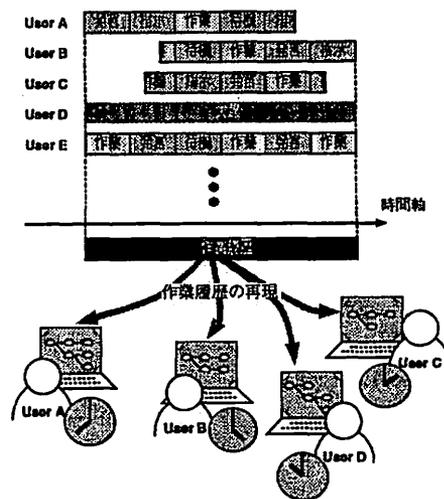


図1: 協調作業と時間の分散化

図1は、先に上げた作業参加者の協調作業が行なわれた過程を時間軸を横にとり示している。User A、User B、User C、User Dがそれぞれ、(1)、(2)、(3)、(4)を意味している。作業終了後、User A～User Dは、他の作業参加者によって進められた作業過程を実際は作業に完全に参加していないが、その作業過程を再現することで、全体を把握することが可能となる。これによって時間の分散化が実現され、離散時間共有が実現される。

2.2 作業の共有

前述した、作業過程を保存することで離散時間作業共有を実現したが、この共有だけではコ

コンピュータを利用するの恩恵は少ないと言える。それは、コンピュータを利用することによる、情報の蓄積や仮想空間における自由な作業実現の機能を十分に活用していないと考えられるためである。

作業の共有では、ある作業の作業から他の作業が触発を受け、新たな作業に導かれ、作業が進められていく。しかし、作業過程のみの離散時間共有においてこの触発を受けた場合は、全体の流れを把握した後、新たに創造を始めることになる。また一方で、作業者は流れに触発を受けることもあるが、ある特定の作業に触発を受けることもある。つまり、作業単位の知的触発へのトリガの集合が単に発生しているのではなく、その作業は何らかの形で再利用され、新たな創造へ導く可能性を包含して発生している。よって、作業単位での再利用機能が必要と考えられる。具体的には、作業は、図形、イメージ、動画と様々な表現によって行なわれるため、一作業の再利用が可能であるならば、これらの内容を新たな作業へ利用することが可能になる。つまり、作業の再利用を行なうことで、ユーザは思考の拡張展開が可能となり、創造支援につながると考えられる。また、この機能の拡張として、コンピュータの仮想空間の機能を応用すると、ある作業の流れの一部を繰り返し再現することが可能である。これは、会議のような会話の流れでは効果はないが、何かを創造する過程を繰り返し再現することで、その創造物がどのように作られていったかを理解する場合に有効である。この作業単位のみならず、作業の流れでの作業共有を実現することが作業効率を向上させると同時に、創造を支援すると考えられる。

3 離散時間協調作業支援システム

前章で述べた離散時間における作業共有を実現するために、統合作業支援プラットフォーム Fusion と、作業単位をイベントオブジェクトとしてネットワーク上で送受信を可能にする ObjectScript について述べる。

3.1 統合作業支援プラットフォーム Fusion

Fusion は、Smalltalk-80 R4.1 の VM (パーティクルマシン) 空間上に構成される。Smalltalk によって、分散オブジェクト環境を実現する上で、重要なクラスライブラリの統一が可能となる。伝統的なクラス体系は、Fusion 内で利用するデータすべてをオブジェクトとして管理を可能にする。また、その環境下で実現されるアプリケーション開発に必要な機能、MOB の体系化が容易に実現可能となる。また、統一されたクラス体系によって、MOB の転送において、オブジェクト自身を転送するのではなく、その生成スクリプトを転送することで、同一のオブジェクトを得る事が可能になる。この Fusion の全体構成を図 2 に示す。

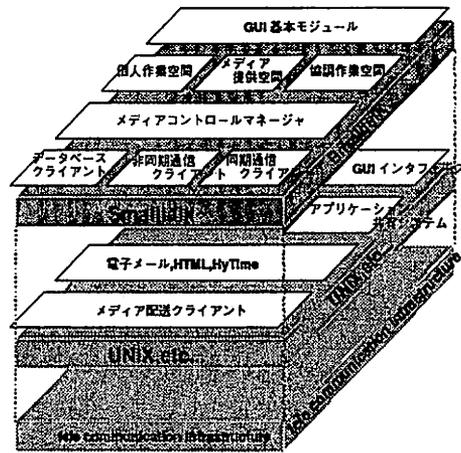


図 2: Fusion の全体構成

Fusion は下位層より、ネットワーク層、OS 層、Smalltalk 層の 3 つに分類される。それぞれの層は、下位層によって、各層に置かれている各種機能が必要とされる、機能補助を行なっている。以下にその各部の機能を説明する。

個人、協調作業空間

個人作業空間と分散環境における協調作業空間において、統合された作業環境をネットワークを介して提供する空間であ

る。個々の空間は作業空間の性質をアプリケーション側でその性質の定義を行なうことで、ユーザに対し個人、協調の空間的な差を意識することなく、作業を進めることが可能にしている。

メディア提供空間

個人の作業情報を逐次保存していき、専用の知識データベースを構成することで、ユーザに必要な様々な MOB の提供を行なう。また、ユーザが行なう作業の新規開始や更新処理の伴う MOB 利用時の処理の自動化の支援も行なっている。

メディアコントロールマネージャ (MCM)

ネットワークを介したアプリケーション間通信、データベースへのアクセス、またローカルな環境におけるアプリケーション間通信を MCM において一元管理を行なう。MCM は各アプリケーションオブジェクトからの MOB 間コミュニケーションのためのメッセージを適切な MOB に伝える作業を行なっている。

データベースクライアント

ユーザからの様々なデータベースサーバへのアクセス制御を管理している。データベースアクセス時に発生する、データベース言語の違いを、このデータベースクライアントで包含し、ユーザに対し検索作業の一般化を行ない、データベース検索を支援する。

同期/非同期通信クライアント

非同期通信クライアントは、電子メール等の時間的同期を必要としないメッセージの送受信機能を管理している。同期通信クライアントは、時間的な管理が必要なメッセージの送受信機能を管理している。両通信クライアントは、下位層に位置するメッセージ配送クライアントと通信することで通信を行なっている。

3.2 ObjectScript

ObjectScript は、ユーザがこのアプリケーションオブジェクトに対して、メッセージ送信を簡単にすることを目的としている。ObjectScript は、宛て先、名前、内容の 3 つから構成されている。宛て先はユーザ ID、名前はアプリケーション ID、内容はアプリケーション ID で指定されたアプリケーションに対してのイベントが記述されている。内容には、イベント以外に転送したい MOB 自身を包含することが可能である。Fusion は、イベントを含む全てが Smalltalk 文で記述されるため、ユーザはメッセージの送信に特別なスクリプトを覚える必要はない。

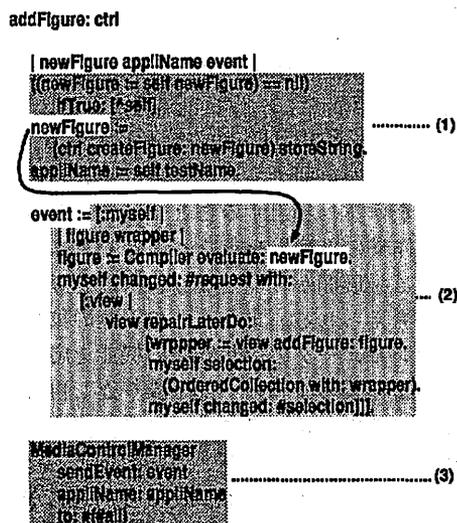


図 3: ObjectScript の例

MCM に対し、図 3 の (3) の部分を実行することでイベントは送信される。図 3 の (2) には、そのイベントの内容がブロック式で記述されており、その内部にある myself には、applName で指定されたアプリケーションオブジェクトが MCM より代入される。(1) で定義された newFigure は (2) のイベント内に包含され、スクリプトの転送先で代入されるアプリケーションオブジェクトに対してのメッセージとして記述

される。これにより、ローカル環境、分散環境でのメッセージ送信の一元化を行なっている。

また、メッセージの記述方法を応用すると、分散環境上の Fusion システムを巡回し、各システムに対してメッセージを送り、その結果を受けとるといような、エージェントへの拡張が可能である。メッセージを Smalltalk 文で記述することにより、エージェントに必要な機能も、すべてアプリケーションオブジェクト、メディアオブジェクトへのメッセージ形式での包含が可能である。

4 離散時間共有会議アプリケーション

今回試作を行なった会議システムは離散時間において会議の流れを再現することと、会議中の発言をオブジェクトとして、再利用を行なうことを目的としている。本章では、構成と実装状況について述べる。

4.1 構成

会議アプリケーションは、発言の表示を管理する流れ図ウインドウ、発言の出力、会議の流れを管理する発言管理部で構成される(図4)。

各ユーザからの発言は一旦発言管理部に渡される。各部の機能を以下に示す。

議題割当時間管理部

会議進行計画から、各議題の重要度に応じて時間を割当て会議を進める。また、議題の進行状況に従って時間の割当てを動的な管理も行なう。会議の進行をスムーズに進めるための指標的な役割をする。

発言出力管理部

キュー内に発言がある場合に起動され、議長によって設定された間隔によってキュー内からの出力管理を行なう。キュー内が空の時は動作を停止し、発言入力部に起動依頼を行なう。

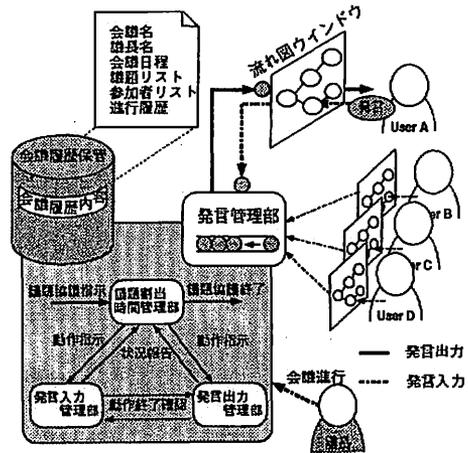


図 4: 会議アプリケーションの構成

発言入力管理部

キュー内为空の時に起動する。設定時間内に発言が入力されるかを監視する。時間内に発言されない場合は、会議参加者に対し発言を促すメッセージを送信し会議の活性化を行なう。発言があると動作を停止し、発言出力管理部に起動依頼を行なう。

発言管理部より一定時間を経て発言が各ユーザに送信され、その内容が流れ図ウインドウに反映される。発言管理部は発言サイクルによって生成される履歴を会議履歴内容として保存を行なう。この履歴には図4に示す会議名、議長名、進行履歴等、会議で利用したすべてのデータが保存され、様々な会議が保存される。会議参加者は会議終了後、この会議履歴内容より任意会議の進行状況を再現することや、履歴の続きより会議を再開することが可能である。

流れ図ウインドウは木構造によって発言の従属性を示している。発言はノード、従属関係はアークを意味している。本論文ではこの従属関係を発言に関係が発生することより発言関係と呼ぶ。参加者からの発言において例を上げると、User A の発言に対して意見を述べる場合はその下位ノードとして生成される。また、会議の流れを明確に示すための支援機能として、ノード

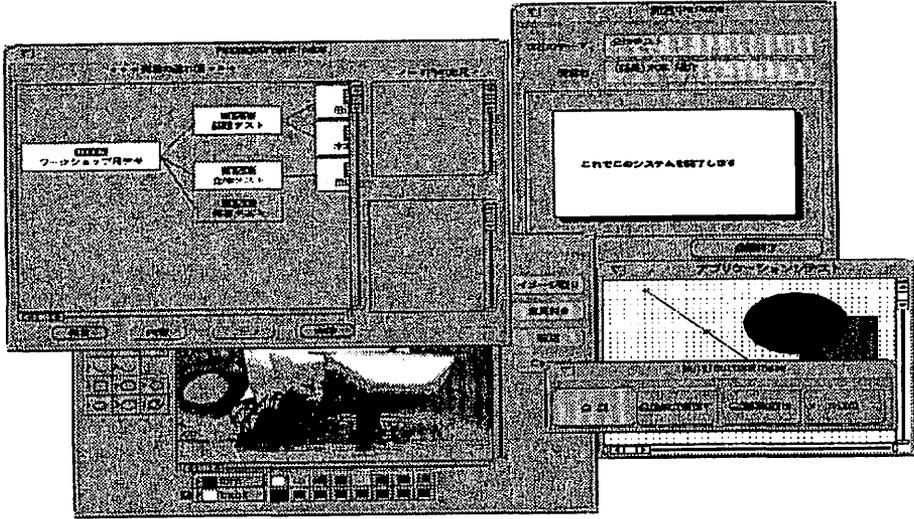


図 5: 会議システムへの実装

の整列、移動機能を提供している。発言ノードは発言内容、発言者名以外に、その発言を生成するための機能を包含したイベントオブジェクトとして保存され、作業状況の保存を行なっている。他の会議参加者はその発言関係とイベントオブジェクトにより会議の流れを知ることとなる。

4.2 実装

会議アプリケーションへの応用を行なった実装状況を図 5 に示す。

会議参加者からの発言が発言管理部より出力されると、発言中ウインドウにその発言内容が出力される。発言が一定時間なされない場合は、発言中ウインドウに発言を促すメッセージの出力が行なわれる。一連の発言は発言関係によって流れ図ウインドウに出力され、発言の流れを示している。また、今話題となっている議題について、何について発言が集中しているかを知るために、ノードの深さ検索機能を提供している。発言には、テキスト、図形、静止画等を含むことが可能であり、これらは全てイベントオブジェクトとして管理される。これらの発言の

流れを管理するために、議長は図 5 に示される以外に議長ウインドウを持つ。議長ウインドウは議題の進行決定、発言の出力間隔を決定している。しかし、議長が行なう作業は全体の流れの調整であり、誰が発言を行なったか、また今、誰が会議に参加しているかはすべて会議アプリケーションで管理しており、議長自身、会議参加者として自由に発言を行なえる。

会議中の発言の流れは、会議終了後、履歴として保存される。この履歴は、前述した会議に同席できなかった会議参加者に対して、会議が行なわれた時間の共有を支援するものである(図 6)。会議履歴を一つ一つ評価していくことで、他の会議参加者が同席せずとも、各参加者が発言を行なっている状況を再現することが可能である。履歴の再現形式は、イベントオブジェクト単位で評価を行なう、イベントオブジェクトを一つ前に戻す、最後まで履歴を進めるがある。会議履歴は任意に評価が行なえるため、繰り返し会議を再現することも可能である。

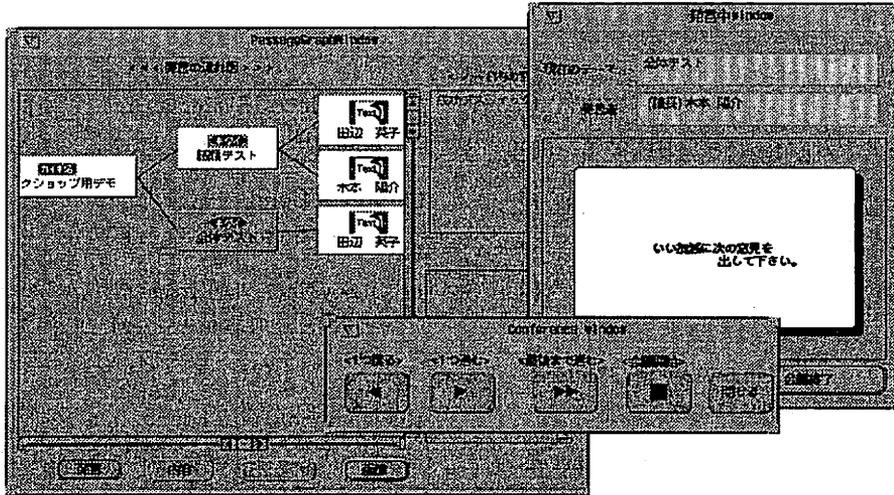


図 6: 会議履歴の再現

5 まとめ

本論文では、協調作業支援における作業共有、時間共有について述べた。その特徴をまとめると以下ようになる。

ユーザの利用環境での特徴

- ユーザからのフォーマル、インフォーマルな情報を包含して、イベントオブジェクトとして伝えることが可能。
- 任意の作業履歴の一部を再現、再利用することが可能。
- 作業履歴によって離散時間における、時間共有、作業共有が可能。

プラットフォームの特徴

- ObjectScript により、スタンドアロンで実現されるアプリケーションは、分散環境においてもその動作が保障される。
- 統合されたクラス体系を利用することにより、MOB 再利用性が向上している。

- MCM を介することでローカル、分散オブジェクトを意識することなく、一般化されたメッセージで MOB 間コミュニケーションを実現することが可能。

今回試作を行なった会議アプリケーションによって、イベントオブジェクトの有用性の確認ができた。本論文で述べた離散時間共有を実現するイベントオブジェクトは、設計図や、プレゼンテーションマネージャのような、作成された内容より、それが作成されるまでの過程を知ることが重要なアプリケーションの共有に大きな効果があると考えている。設計の過程の一部を取り出し、別の作業に反映することで、作業そのものの再利用が可能となる。今後の課題として、イベントオブジェクトの具体的な作業への応用を行うアプリケーションの開発を行なう予定である。

謝辞

本研究に御支援をいただいた、国際メディア研究財団殿、(株)PFU 殿に深く感謝いたします。

参考文献

- [1] 木本陽介, 新井敏正, 服部進実. 状態遷移会議モデルによるマルチメディア分散協調作業支援システム, 信学技法, HC 93-87, (1994,3).
- [2] 木本陽介, 鴨宮淳, 川畑豊, 服部進実. オブジェクト指向ヒューマンコンピューティングによる知識ナビゲーションの一考察, 情報処理, DPS 68-9, (1995, 1).
- [3] HotJava browser and Java language
< <http://java.sun.com> >.
- [4] Spring distributed operating system
< <http://www.sun.com/cgi-bin/show?tech/projects/spring/index.body> >.
- [5] 岡村英明, 石川裕, 所真理雄. リフレクション機能を持つ分散プログラミング言語システム AL-1/D, 日本ソフトウェア科学会 WOOC'92.
- [6] 伊東達雄, 今城広志, 岡野浩三, 東野輝夫, 松浦敏雄, 谷口健一. グループワークを考慮した協調計算システムにおける動作プログラム群の生成と分散実行, 情報処理学会論文誌 (June 1995).
- [7] 松下温, 岡田謙一, 勝山恒男, 西村孝, 山上俊彦. 知的触発に向かう情報社会 共立出版 (April 1994).
- [8] R. Macneil. Generating Multimedia Presentations Automatically using TYRO, the Constraint, Case-Based Designer's Apprentice, Proc. of IEEE Visible Language Workshop (October 1991).
- [9] 湯浅敬, プラートイブ K. シンハ. ネットワークマルチメディアアプリケーションのためのオブジェクトモデル, 日本ソフトウェア科学会 WOOC'92.
- [10] 田中 哲男, 神野 俊昭. 分散オブジェクト環境におけるメディアブラウザの開発, 情報処理学会研究会 FI-31-2, 1993.
- [11] 川村湯真. オブジェクト指向コンピュータを作る, アスキー.
- [12] ParcPlace Systems Inc. ObjectWorks Smalltalk User's Guide.
- [13] Adele Goldberg, David Robinson, 及川一成他訳. Smalltalk-80 —言語詳解—.
- [14] J. Boykin, D. Kirschen, A. Langerman, S. LoVerso, 寺本信一訳. Mach オペレーティングシステム, トップラン.
- [15] Open Software Foundation 著, (株) 日立製作所 ソフトウェア開発部訳. OSF DCE 入門, トップラン.
- [16] M. Bryan 著, 山崎俊一監訳, 福島誠訳, SGML 入門, アスキー出版.
- [17] Laura Lemay 著, 武舎広幸, 久野禎子, 久野靖訳. HTML 入門 (WWW ページの作成と公開), プレンティスホール.
- [18] H. Masuhara, S. Matsuoka, T. Watanabe and A. Yokozawa. Object-Oriented Concurrent Reflective Language can be Implemented Efficiently. Proc. of OOPSLA'92 (October 1992).
- [19] 千葉 滋 他. Object Community: 自己反映計算を用いたグループウェアの為にデータ共有機構, WOOC'92 オブジェクト指向コンピューティング I, 近代科学社.