仮想世界データベースシステムにおける共有型作業環境のための仮想世界同期法

Synchronization Method of Shared Work Environment in VDWB2: A Virtual World Database System

渡辺 知恵美♥　　増永 良文♠

Chiemi WATANABE　Yoshifumi MASUNAGA

仮想世界データベースシステム VWDB2 はデータベース機能を完全に備え持つ拡張型ネットワークバーチャルリアリティシステムである．本論文では，仮想世界データベースシステム VWDB2 上に構築された共有型作業環境のための仮想世界同期法の提案を行う．VWDB2 では複数の VR クライアントと 1 台のバックエンド DB サーバによる集中型管理方式を用いており，共有型作業環境の特徴を考慮した VWDB トランザクションを用いて各 VR クライアントからの更新要求及び問合せを一元管理する．この方法を用いた場合，共有される仮想世界の一貫性は保証されるが，スケーラビリティ及び同期性の向上に伴ってサーバにアクセスが集中し全体のパフォーマンスが低下してしまう恐れがある．そこで我々はゴーストオブジェクトという概念を導入することによって，共有型作業環境として適切なスケーラビリティと同期性を保ちながらサーバへのアクセス集中を軽減するための仮想世界同期法を提案し実装した．

The VWDB2 prototype system was built based on the network virtual reality system architecture, where several front-end virtual reality systems and one back-end object-oriented database system are system-integrated via a high-speed computer network. In this paper, we propose a novel synchronization method for shared work environment realized on VWDB2. Because each update or query request for virtual objects is treated as a transaction to the back-end database system, VWDB2 can guarantee the consistency of the shared work environment. However, when its scalability and synchronism becomes high, the number of accesses from VR clients becomes concentrated to the back-end database system, which tends to decrease the performance of the whole system. In order to resolve this problem, we have introduced the "neighboring ghost objects" which can decrease the number of accesses to the back-end server while keeping the adequate scalability and synchronism.

1. はじめに

近年，バーチャルリアリティ(VR)システムの実用化に伴い，VRシステムにおけるデータベース機能の必要性が認識されつつある．実際に，ウォークスルーに伴う検索機能[4]や仮想世界オブジェクトの永続的管理機能[3]なども開発されてきている．これらのシステムはアプリケーションの用途に合わせてデータベース機能の一部を利用しているが，データベースシステムはVRシステムをサポートするために必要と考えられる有用な機能を数多く備えており，これらのデータベース機能を完全に備え持つVRシステムを開発することは大変意義あることと考えられる． 我々はこのようなバーチャルリアリティをデータベースにしたシステムの実現を目指し，VRシステムとDBシステムをシステム連携したVirtual World Database System (VWDB)の開発を進めてきた[6][7].

VWDBは当初一台のVRシステムと一台のデータベースシステムをシステム連携してプロトタイピングが行われたが，最新のバージョンでは複数台のVRシステムと一台のデータベースシステムを統合してプロトタイプが構築されている．このような構成のVWDB2は，共有型作業環境(Shared Work Environment)の支援に有効に働くと考えられる．ここに，共有型作業環境とは，複数のVRシステムによるネットワークVR (NVR)システム[9]によって共有化された仮想世界オブジェクトを,同じ空間に没入した複数のユーザが共同で編集出来る環境である．これらは，従来のNVRシステムで実現されるシミュレーションシステムやチャットシステムに比べて規模は小さく（2～数十人），共有する仮想世界オブジェクトの管理が重要となる．しかし，共有型作業環境におけるオブジェクト管理機能の研究は未だ初期段階にあり，加えて多数の作業の同時実行や障害時回復機能の実現を考えると，共有型作業環境の実現技術はまだ確立されているとは言いがたい．そこで，我々は文献[11]にて，VWDB2上でのデータベース機能を根底にした共有型作業環境の実現を目指し,仮想世界の一貫性を管理するためのVWDBトランザクションを導入した.

本論文では，文献[11]にて提案されたVWDBトランザクションで仮想世界の一貫性を高く保持しつつ，仮想世界のスケーラビリティと同期性を保つための仮想世界同期法の提案を行う.第2章でVWDB2における共有型作業環境のシステム構成と文献[11]で提案したトランザクション概念について概略を述べた後，第3章にて同期性が低下した場合に起こりうる問題点を明確にし，それを回避するためにゴーストオブジェクトという概念を導入した同期法を提案する.

2. 仮想世界データベースシステム：VWDB2における共有型作業環境

VWDB2は，複数台のVRクライアントとバックエンドのオブジェクト指向データベースシステムによる仮想世界集中管理方式[8]で共有型作業環境を実現している(図1)．VRクライアントはJAVA言語用の仮想世界構築APIであるJAVA3Dを用いて実装し，バックエンドDBは商用のオブジェクト指向DBであるエクセロン社のObjectStoreを利用しJAVA言語で実装している．クライアントとサーバ間の通信プログラムはHORB[9]を用い，マルチキャストはHORB extentionメッセージングサービスを用いている.

VWDB2でVRクライアントのアプリケーションを起動すると，VR クライアントはデータベース中にある仮想世界のコピーをローカルに作成し，VR システム上に表示する．そして，各VR クライアントの持つ仮想世界とデータベース中に格納されている仮想世界との同期を取ることで全てのクライアントが「同じ仮想世界を共有している」感覚でVR シ

♥ 学生会員　お茶の水女子大学大学院人間文化研究科博士後期課程　chiemi@dblab.is.ocha.ac.jp

♠ 正会員　お茶の水女子大学理学部情報科学科 masunaga@is.ocha.ac.jp

ステムを利用することができる．

VWDB2 では仮想世界における問合せや更新要求はすべて VWDB2 トランザクションとしてバックエンドサーバで処理される．我々は文献[11]にて定義した VWDB トランザクション概念を基に，データベースの一貫性保障や同一の仮想空間に没入した複数のユーザ間の同時実行制御を行う．VWDB トランザクションでは，図 2 に示すような移動トランザクションのようにインタラクションがある時間間隔で（不定回）複数連続して実行されることによって一連の意味のある操作が行われるとき，これを連続操作トランザクションという 2 階層のトランザクションとして管理する．例えば移動トランザクションは，移動の開始から終了までのトップトランザクションと，ある時間間隔で実行される位置更新サブトランザクションとで構成される連続操作トランザクションである．VWDB トランザクションは，共有型作業環境の特徴を考慮し緩和された ACID 特性を保証する．例えば，「操作の途中経過をできるだけリアルタイムに他ユーザに公開する必要があるという」性質を考慮して分離性を緩和し，サブトランザクションがコミットする毎に他クライアントにサブトランザクションの結果をマルチキャストさせている．

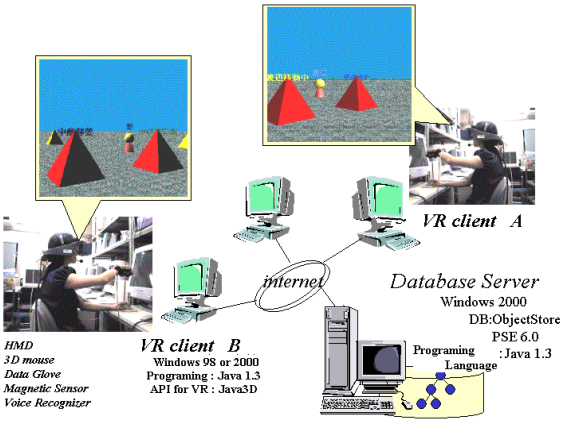

図 1　VWDB2 のシステム構成

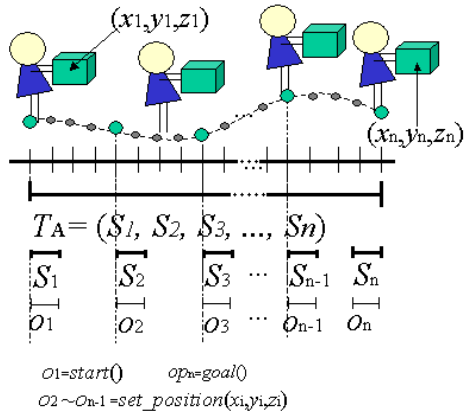

図 2　移動トランザクション

3. 共有型作業環境における仮想世界同期法

共有型作業環境では，仮想世界での同時作業人数（スケーラビリティ）及び仮想世界の同期間隔（同期性）が重要となる[1]. VWDB2 では連続操作トランザクションのサブトランザクションがコミットされた時点で他クライアントへのマルチキャストによって同期が行われるため，同期間隔とはすなわちサブトランザクションの発行間隔となり，発行間隔が短くなるほど同期性が高い．

共有型作業環境ではスケーラビリティ及び同期性ともに高いほど良いが，我々が採用している仮想世界集中管理方式ではバックエンドデータベースサーバの負荷が高くなり全体のパフォーマンスが低下するという問題がある．VWDB2 は同時作業人数が 2 人～数十人という比較的小規模なアプリケーションを対象としているが，例えば 10 人同時にオブジェクトを移動した場合，サブトランザクション発行間隔が 10 ミリ秒ならば 1 秒間に 1000 ものアクセスが集中することになる．

そこで本章では「ゴーストオブジェクト」という概念を導入することによって，適切な同期性を保ちながらサブトランザクション発行間隔を（描画サイクルと比べて）長くし，アクセス集中を軽減する仮想世界同期法を提案する．

3.1 同期性の低下に伴う問題点

まず同期性の低下に伴って生じる問題点を明らかにする．図 3 は仮想共同作業空間上でクライアント A とクライアント B が各々同時にオブジェクト O_A とオブジェクト O_B を移動するトランザクション T_A，T_B を実行している様子である．各トランザクション T_A,T_B では発行間隔 a でサブトランザクションが発行される．このとき発行間隔が広がることによって以下のような現象が起きる場合がある．

現象 1: T_A の位置更新サブトランザクションはある時間ごとに発行され，他のクライアントにマルチキャストされるため，クライアント C_B 上ではオブジェクト O_A が急に動くように感じられる．

現象 2:オブジェクト間の衝突検出はバックエンドデータベースで行われるため，クライアント側で起こったはずの衝突が検出されない場合がある．（図 3）

現象 3:現象 2 と同じ理由で，クライアント側では起こっていないはずの衝突がバックエンドデータベース側で検出されてしまう場合がある（図 4）．

ここでは特に現象 2 と現象 3 を図 3 と図 4 を用いて説明する．まず，現象 2 では，ユーザ A はオブジェクト O_A を移動し，時刻 t_i に地点 P_1 で S_i^A を発行し，時刻 t_i+a に地点 P_3 で S_{i+1}^A を発行している．またユーザ B はオブジェクト O_B を移動し時刻 t_j に地点 P_2 で S_j^B を発行している．ユーザ B は更に移動を続け，時刻 t_j+b(b<a，t_i+a=t_j+b)には地点 P_3 まで移動させている．この時点でクライアント B 上の仮想世界では，オブジェクト O_A と O_B が衝突しているように見える．しかし，この時刻では，オブジェクト O_B のサブトランザクションは発行されておらず，データベース上では衝突していないため，そのまま O_B はオブジェクト O_A を通り抜ける結果となる．また現象 3 では，ユーザ A はオブジェクト O_A を図 3 の右上方向に移動し，時刻 t_i に地点 P_1 への位置更新サブトランザクション S_i^A を，さらに時刻 t_i+a に地点 P_3 への位置更新サブトランザクション S_{i+1}^A を発行している．そのあと，A は O_A を移動しつづけ，時刻 t_1+a+b（b<a）には地点 P_4 まで持っていったとする．一方，ユーザ B はオブジェクト O_B を図 4 の左上方向に移動し，時刻 t_i+b に地点 P_2 への位置更新サブトランザクション S_i^B を，さらに時刻 t_i+a+b に地点 P_3 への位置更新サブトランザクション S_i^B を発行している．このとき，クライアント C_A ではユーザ A が O_A を P_4 まで移

動させていたのにもかかわらず，地点 P_3 での衝突計算が行われ，ユーザ A が意図しない方向に移動してしまう．

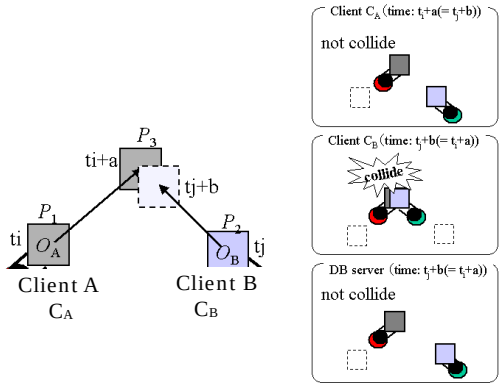

図 3．現象 2

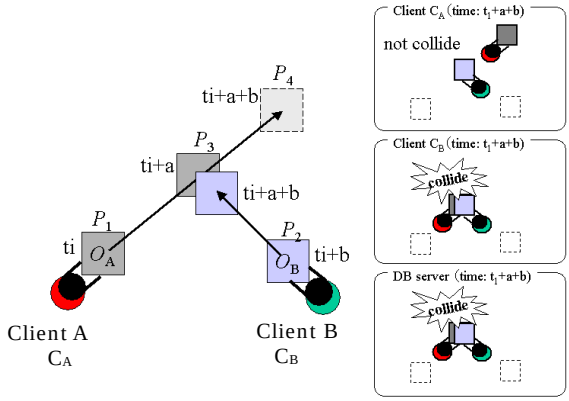

図 4．現象 3

3.2 ゴーストオブジェクトの導入

前節の現象は，各連続操作トランザクションのサブトランザクションが一定間隔（前節では a）毎に発行されることによって起こる．つまりサブトランザクションが発行された時刻 t から次に発行される時刻 t+a までクライアント側のオブジェクトの位置とデータベース及び他クライアント側のオブジェクトの位置が同期されていないことが原因となる．この矛盾をなくすために表示をどちらかの状態に統一する方法もある．しかし，各クライアントでサブトランザクション間に実行されるローカル操作は描画サイクル毎（約 50ms）であり，データベースへ発行するサブトランザクションの間隔をそこまで狭めるのには限界がある．また，データベースでの状態を統一するためにサブトランザクション間に実行されるローカル操作を止めてしまうと，対話性が極端に落ちてしまう．そこで，我々は各クライアントのアプリケーションに「ゴーストオブジェクト」を導入することによってその解決を図った．

3.2.1 ゴーストオブジェクト

クライアントで連続操作トランザクションを実行しているとき，サブトランザクション間でローカルでのオブジェクトの状態とデータベース上でのオブジェクトの状態とが異なっている場合，これらを両方表示することとした．この場合，データベース上でのオブジェクトの状態を表示したものを「オリジナルオブジェクト」，クライアント上のオブジェクトの状態を表示したものを「ゴーストオブジェクト」とする．移動トランザクションにおけるゴーストオブジェクトの例を図 5 に示す．サブトランザクション発行間隔を 500ms とし，VR クライアントでの描画サイクルを 50ms 間隔とする．また，クライアント上で行われる処理の時間は p(p<50ms)，共有型作業環境における操作の処理時間は q(50<q<500)であるとする．この場合，オブジェクト O_A の移動トランザクション T_A を開始すると，ゴーストオブジェクト O_A'が生成される．ユーザは基本的にこのゴーストオブジェクト O_A'を移動させることになる．時刻 t_0 に開始サブトランザクション S_1 が発行された後，位置更新サブトランザクション S_2 が実行されるまでの間，50ms 間隔でゴーストオブジェクト O_A'の位置更新操作が実行される．これは，VR クライアントのみで完結する操作である．そして，時刻 t_0+500(ms)に S_2 が実行され，ゴーストオブジェクトの位置である P_2 地点にオリジナルオブジェクトを移動させるよう位置更新サブトランザクションを発行する．これにより，オリジナルオブジェクトは時刻 t_0+500+q に p_2 地点に移動する．ゴーストオブジェクトとオリジナルオブジェクトは線でつながれ，ゴーストオブジェクトを使ってオリジナルオブジェクトを引っ張る感覚で移動を行うことができる．

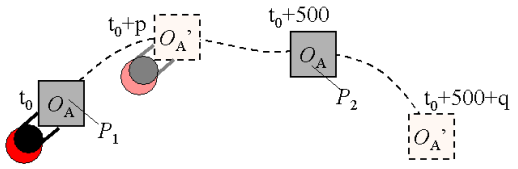

図 5．ゴーストオブジェクト

3.2.2 ゴーストオブジェクトの共有

ゴーストオブジェクトを用いて 4.1 節で挙げた現象を緩和させるために，我々は近隣で連続トランザクションを実行しているクライアント同士が互いのゴーストオブジェクトを共有させる方法を用いる．例えば，クライアント C_A，C_B，C_Cから各々ユーザ A，B，C がログインしてオブジェクト O_A，O_B，O_Cの移動トランザクションを実行し，現在，データベース上の仮想世界が図 6(a)に示す状態になっているとする．このとき，次のサブトランザクションまでの移動可能距離を r とし，操作中のオブジェクトを中心とした半径 r の球状の範囲をオブジェクトの「近隣」であると考える．図 6(a)では O_A と O_B，O_A と O_C がそれぞれ近隣となる．このとき，O_A の移動トランザクション T_A を発行しているクライアント C_A と，O_B の移動トランザクション T_B を発行しているクライアント C_B との間に P2P のネットワーク通信路を確保し，O_A のゴーストオブジェクト O_A'と O_B のゴーストオブジェクト O_B'を共有する(図 6(b))．なお，P2P 通信で共有するゴーストオブジェクトはあくまでオリジナルオブジェクトに対する仮の姿なので，オリジナルオブジェクトに比べて一貫性を厳密に確保する必要性は低いため，楽観的な一貫性管理手法[13]を取り入れる．

3.2.3 ゴーストオブジェクトの共有による解決策

前節で述べたようにゴーストオブジェクトを近隣にいるクライアント同士で共有することによって，4.1 節で挙げた現象を最小限に抑えることが出来る．まず現象 1 は，近隣オブジェクトのゴーストを他のクライアントが見られるようにすることで，次のサブトランザクションで更新されるオブジェクトの状態を予測することが出来るため，解消される．

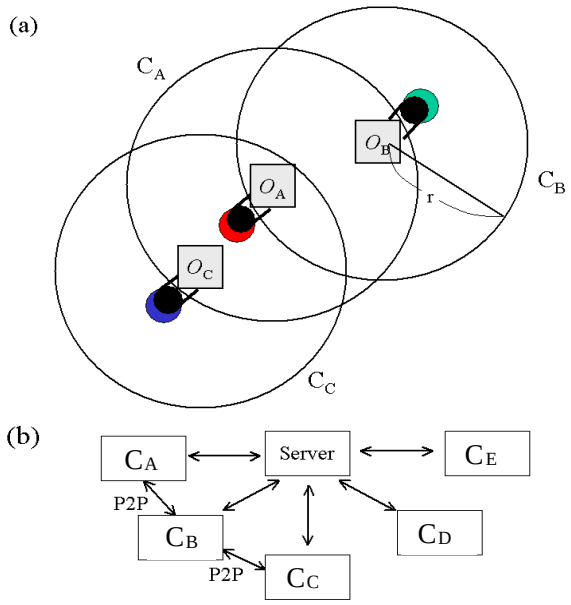

図6. 近隣でのゴーストオブジェクトの共有

現象 2 は以下のように解消する：オブジェクト O_A と O_B が近隣にあり，C_A と C_B が P2P で接続され，ゴーストオブジェクト O_A'と O_B'が共有されているとする．ゴーストオブジェクト O_A'と O_B'が衝突したときの制御フローを図 7 に示す．C_A と C_B の移動トランザクション T_A と T_B は，通常では一定間隔（ここでは a）でサブトランザクションを発行するが，ゴーストオブジェクト O_A'と O_B'の衝突がクライアント上で検出された時点（C_B は O_B'を更新したとき，C_A は O_B'更新を受信したとき）で，特別に各々サブトランザクションを発行する．これにより，バックエンドデータベース側でオリジナルオブジェクト O_A と O_B の衝突が検出されることとなる．また，各クライアントではサブトランザクション発行後，ゴーストオブジェクト同士の衝突計算を行い，各々ゴーストオブジェクト O_A'と O_B'の位置を更新する．その結果，各クライアントでは，共有されたゴーストオブジェクトの衝突，及びオリジナルオブジェクトの衝突が実行されることになり，現象 2 は解決される．

また現象 3 については，サーバ側で衝突が検出されたときにクライアントでの衝突検出の通知（図 7 の ACK）がない場合はその衝突を無効にすることによって解決することが出来る．

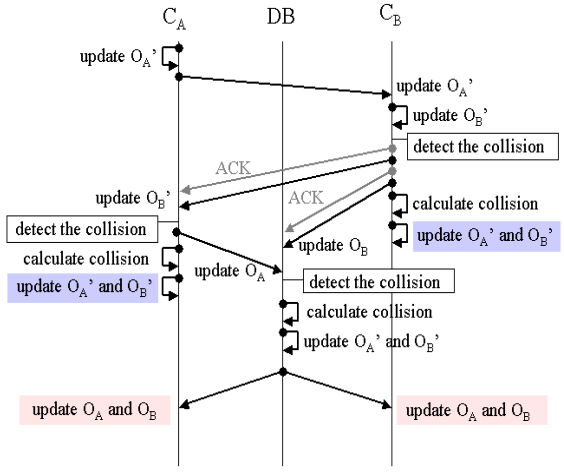

図 7.ゴーストオブジェクト衝突時の制御フロー

4. まとめと今後の課題

VWDB 上で実現する共有型作業環境が適切なスケーラビリティと同期性を保つために，ゴーストオブジェクトを用いた仮想世界同期法を提案した．今後の課題としては今回提案したゴーストオブジェクトによる仮想世界同期法を用いてどの程度のスケーラビリティと同期性を保つことができるかの評価を行う予定である．

[文献]

[1] Meehan, M.: “Survey of Multi-User Distributed Virtual Environment”, in Course Notes: Developing Shared Virtual Environments, ACM Press (1999).
[2] Calvin, J., Dickens, A., Gaines, B., Metzger, P., Miller, D. and Owen D.: “The Simnet Virtual World Architecture”, Proceedings of IEEE VRAIS, pp.450-455 (1993).
[3] Schnabel, M. A. and Kvan, T.: “Implementing The First Virtual Environment Design Studio: Architectural Education for the Asian Century”, Proceedings of the 1st ACAE Conference on Architectural Education, pp. 157-166 (2001).
[4] Broll, W., Meier, E., Schardt, T.: “The Virtual Round Table - a Collaborative Multi-User Environment”, In Proc. of the ACM Collaborative Virtual Environments 2000 (CVE2000), pp. 39-46 (2000).
[5] Galli, R., Luo Y., Mu3D: “A Causal Consistency Protocol for a Collaborative VRML Editor”, Proceedings of Fifth Symposium on the Virtual Reality Modeling Language (VRML2000), pp.53 – 62 (2000).
[6] Masunaga, Y. and Watanabe, C.: “The Virtual World Database System –Its Concept, Design and Prototyping-”, Advances in Multimedia and Databases for the New Century -A Swiss/Japanese Perspective-, pp.61-70, World Scientific (2000).
[7] Masunaga, Y. and Watanabe, C.: “Design and Implementation of a Multi-modal User Interface of the Virtual World Database System (VWDB)”, Proceedings of Seventh International Conference on Database Systems for Advanced Application (DASFAA'01), pp.294-301, IEEE CS Press (2001).
[8] Funkhouser, T. A.: “Network Topologies for Scalable Multi-User Virtual Envionment”, Proceedings of VRAIS, pp.222-228 (1996).
[9] HORB home page, http://www.horb.org
[10] Rusinkiewicz, M. and Sheth, A.: “Specification and Execution of Transactional Workflows”, in Modern Database System (eds. W. Kim), pp.592-620 (1995).
[11]渡辺知恵美，大杉あゆみ，佐藤こず恵，増永良文：“仮想世界データベースシステムにおける共有型作業環境のためのトランザクション概念の導入”，情報処理学会論文誌：データベース（TOD），Vol.42, No.SIG15 (2002) (to appear).

渡辺 知恵美　Chiemi WATANABE
お茶の水女子大学大学院人間文化研究科博士後期課程在学中．2000 年お茶の水女子大学大学院人間文化研究科博士前期課程修了．仮想世界データベースシステムの研究・開発に従事．情報処理学会学生会員．日本データベース学会学生会員．

増永 良文　Yoshifumi MASUNAGA
お茶の水女子大学理学部情報科学科教授．1970 東北大学大学院工学研究科電気及通信工学専攻博士課程修了，工学博士．データベースシステムの研究・開発に従事．情報処理学会データベースシステム研究会主査，ACM SIGMOD 日本支部長，情報処理学会監事などを歴任．情報処理学会フェロー．電子情報通信学会フェロー．日本データベース学会副会長．IEEE-CS，ACM 各会員．著書に「リレーショナルデータベースの基礎―データモデル編―」(オーム社)，「リレーショナルデータベース入門」(サイエンス社)，監訳本に「オブジェクト指向データベース入門」(Won Kim 著，共立出版)など．