教官の研究教育活動等報告書データベースシステムの開発と運用

野中　裕介[†]　井上　創造[†]　秦野　克彦[†]　原田　努[†]
乃村　能成[††]　岩井原瑞穂[††]　峯　恒憲[††]　牛島　和夫[††]

Development and Operation of a Document Database for University Research and Education Activities

Yusuke NONAKA[†], Sozo INOUE[†], Katsuhiko HATANO[†], Tsutomu HARADA[†], Yoshinari NOMURA[††], Mizuho IWAIHARA[††], Tsunenori MINE[††], and Kazuo USHIJIMA[††]

あらまし　九州大学教官の研究教育活動等報告書データベースは，各教官の研究教育活動の報告書に関して，インターネットを通じた各教官による報告書の入力や更新，管理者による報告書の検査等の管理作業，閲覧者による報告書の検索や閲覧を行うためのシステムである．教官が入力を行うための方法は，複数の方法からの選択が可能である．また，報告書は各教官ごとに異なったフォーマットを選択することが可能で，データベース構造にリレーショナルデータベースとSGMLを混在させることでその機能を実現している．システム設計や運用体制についての評価をシステムのログや開発者に送られた電子メールに基づいて行った結果，システム運用時における管理者を複数設けて分担する分散管理による負担軽減が機能していることや，システム更新時の修正箇所が複数のモジュールにわたることが少ないことが確認できた．

キーワード　情報検索ソフトウェア，SGML，ソフトウェアプロセス

1. まえがき

九州大学教官の研究教育活動等報告書データベース（以下，本システム）は，九州大学教官の研究業績や社会連携活動などからなる「研究教育活動等報告書（以下，報告書）」をデータベース化したものである．これは大学人として，研究教育の自己評価を報告書としてまとめインターネットで公開して，九大の教官の活動ぶりを広く社会に認識してもらうという，社会に公開された大学を目指す試みである．

筆者らを含む，教官4名，学生5名の構成によるグループで，本システムの開発は始まった．1997年の開始以来，1998年の公開，そして改訂を重ねて現在の版に至っている．本システムは，学内外から多数の利用がされている[4]．

本論文では，以下，2.では，開発の目的，すなわち，大学における研究教育紹介のためになぜこのようなシステムが必要となったのか，を述べる．従来のシステムになく，実現すべき機能であった，教官自身の手によるデータの随時更新，専門分野の違いによる多様な研究教育活動についての記述を可能とすること，多種多様な入力方法，報告書の検査機構，についての詳細を述べる．3.では，本システムの概要や構成について，特徴的な点を中心に説明を行い，4.では，開発や運用のプロセスを説明する．5.では，3.，4.の内容を受けて，システムログや開発者に寄せられたメールといった運用の記録から，設計や運用のプロセスについての評価を行う．6.では，同様の目的をもつシステム，及びソフトウェア開発プロセスの分析という二つの視点から関連研究との比較を行う．

2. 開発の目的

本システムは，教官の研究教育活動等報告書データ

[†] 九州大学大学院システム情報科学府，福岡市
Graduate School of Information Science and Electrical Engineering, Kyushu University, 6–10–1 Hakozaki, Higashi-ku, Fukuoka-shi, 812–8581 Japan
[††] 九州大学大学院システム情報科学研究院，福岡市
Faculty of Information Science and Electrical Engineering, Kyushu University, 6–10–1 Hakozaki, Higashi-ku, Fukuoka-shi, 812–8581 Japan

ベース化ワーキンググループ（筆者の１人を含む）により要求仕様案の作成が行われ，九州大学自己点検評価委員会の議を経て最終的な要求仕様が決定された．

本システムの開発の目的は，従来の教官の業績紹介では，論文リストのみといったものになりがちなのに対し，教官の研究テーマに対する考え方，教育に対する考え方，更に社会との連携活動，医系学部の診療活動といった，これまでに紹介されることのなかった社会への貢献について，教官自ら自由作文で語る機会を作るものである．

日本全国の大学において，研究者のデータベースとして公開されているのは所属や専門分野，論文リストといった簡単な内容のものばかりである．これに対し本システムによるデータベースは，自由作文を含んだ長文の研究者情報を提供している．また教官がオンラインでデータを随時更新できるリアルタイム性も，開発時の日本の大学では提供されていなかった．従来の研究者データベースでは，１年に一度の更新を行うために書面でデータを集め，それを半年かけてデータベースに入力するため，２年くらい古い情報が出てくることがあり，転出した教官がいつまでも載っているといった問題があった．

報告書の項目については，各学部や大学院における研究教育活動は多様であり，一つの統一的な評価基準を採用するのは困難と思われた．例えば，文系学部では論文のほかにも評論や翻訳といった活動があり，更に地域社会への様々な貢献がある．理系学部のように論文リストだけというわけにはいかない．そこで，全学で共通で使用できるような項目を列挙した．更に各項目ごとに文字数制限を設けることにした．自由作文の項目を多くし，自分の好きなスタイルで記述してもらえるようにするが，あまりに長文の報告書は読者に苦労を強いるものであるため，文字数制限があった方がよい．

各部局（学部や大学院，センター）において，研究教育活動を記述する項目は最適なものはそれぞれ異なっていてもおかしくないであろうから，各部局ごとに異なる報告書の様式を採用できるようにしなければならない．ソフトウェアの開発においては，文書の様式定義を各部局ごとに変えられるようにして対処している．また年度ごとに様式の変更があったとしても，過去の報告書が読めなくなるようでは困るので，異なる様式が混在していても，検索や入力が可能でなければならない．現在までのところは最初の様式一つで本システムは運用されている．翻訳について件数を記入する欄が欲しいなど，様式に対する要望がいくつか寄せられており，様式も検討の後変更されるであろう．

本システム開発でのもう一つの大きな課題は，データの入力方法である．報告書は教官自身が作成して，手元のパソコンからアクセスして報告書をアップロードしてもらうようにしたい．これで従来の報告書のような紙面のやり取りが不要になる．また随時報告書を更新してもらうことで，報告書を最新の状態に保てるようになる．冊子の報告書だと原稿を収集して，一度校正して，印刷して半年後にようやく報告書が配付できるが，次の年まで１年は更新されないため内容が陳腐化してしまう．オンラインで入力できればよいのであるが，すべての利用者が電子メールや Web ブラウザの操作に慣れていることは，期待できない状況であった．そのため，報告書の入力/提出方法は多種多様のものを用意して，その中から選んでもらえるようにした．Web ブラウザが使える場合は Web ブラウザで，ワープロで文書の作成を行っている場合はフロッピーでの提出で，更に書面でも提出できるように，記入用紙をプリンタから出力できるようにした．この場合はデータ入力を行ってもらう人が別に必要となり，もはやオンライン方式ではないが，やむを得ないだろう．

もう一つ重要な問題となったのは，報告書に書かれるであろう内容である．自由作文において，報告書の趣旨から大きく外れた誹謗(ひぼう)中傷や事実に反することが書かれる可能性が否定できない．安全策として，各部局で報告書の内容をチェックして，問題がない報告書のみを公開するような仕組みを入れるようにした．もし報告書が提出されれば自動的に公開されるようなオンライン入力システムでは，それを悪用するケースがないとも限らない．問題があると思われる記述は，筆者に通知して再考を求めると同時に担当する委員会で検討するというシステムを置いて，万全を期することにした．

検査をする人を各部局で配置してもらうという仕組みを置くことにして，更に各部局でのデータ入力の補助をする仕事として，データ入力の催促や教官の転出/転入/退職，部局間の異動といった人事異動も，各部局で独立して管理できるようにした．またデータ入力の方法も，各部局ごとに指導してもらえれば，データベースの管理者が質問の対応に忙殺されずにすむであろう．実際のところ，各部局の担当者に仕事の負荷がどれくらいかかるかは，システム設計段階では見積も

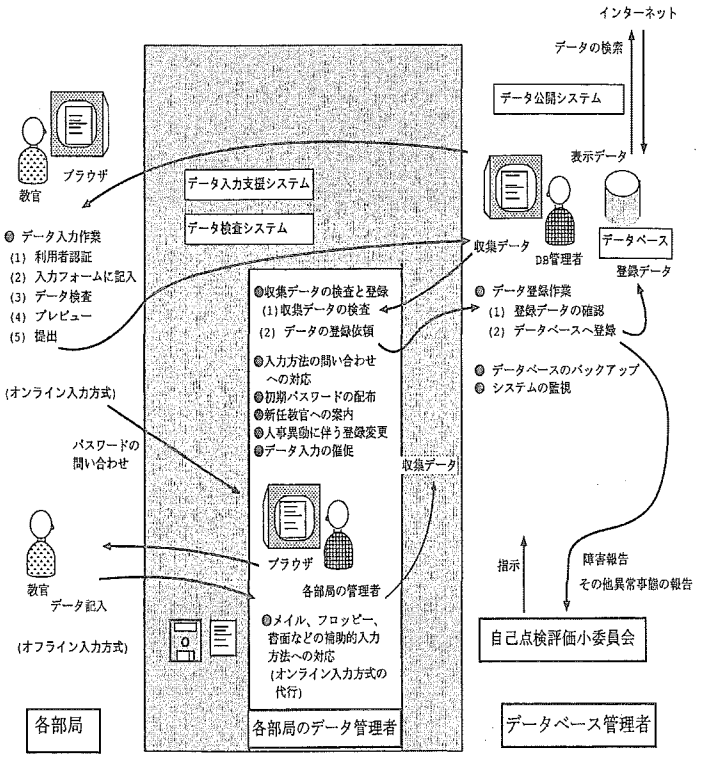


図 1　システム構成及び作業の分担
Fig. 1　Overview of the system structure and share of work.

りにくいところがあるため，システムが稼動を始めてから，過度に作業が集中している担当者の負担を減らすように，入力作業補助の役割分担を適宜見直さなければならないであろう．システム構成及び作業の分担の概略を図 1 に示す．

3. システム構成

3.1 システムの概要

本システムは，報告書を主要なデータとするデータベースの入力，公開，保守に使用される．入力，公開，保守のためのインタフェースとしては，主として HTTP (Hyper Text Transfer Protocol) が用いられるため，Web ブラウザ (以下，ブラウザ) がユーザインタフェースとなる．

3.1.1 利用者の分類

本システムの利用者は，以下のように分類される．

[閲覧者] 公開された報告書を閲覧する者．

[教官] 報告書の入力を行う者．

[部局データ管理者] ある部局の一部または全部の教官についての管理を担当して，報告書の検査や教官アカウントの管理，代行入力などを行う者．各部局に 1 名以上．

[データベース管理者] データベースの保守を行う者．システム (全学) に 1 名．

閲覧者以外にはアカウントが発行されて，利用者認証が行われる．

3.1.2 データベースの入力

入力方法は，各部局ごとのデータ入力方針，あるいは，各教官にとって最も効率の良い作業方法によって選択することができるいくつかの方法がある．

a) 教官による直接入力

教官は，クライアントの計算機上でブラウザを利用して入力欄に報告書の内容を記述した後にサーバに送信することで報告書を提出する．その教官を担当している部局データ管理者は，提出された報告書を検査した後，**3.1.3** の方法で公開する．

本システムでは長文を入力するため，入力中にブラウザを不用意に終了してしまうことなどによるデータ

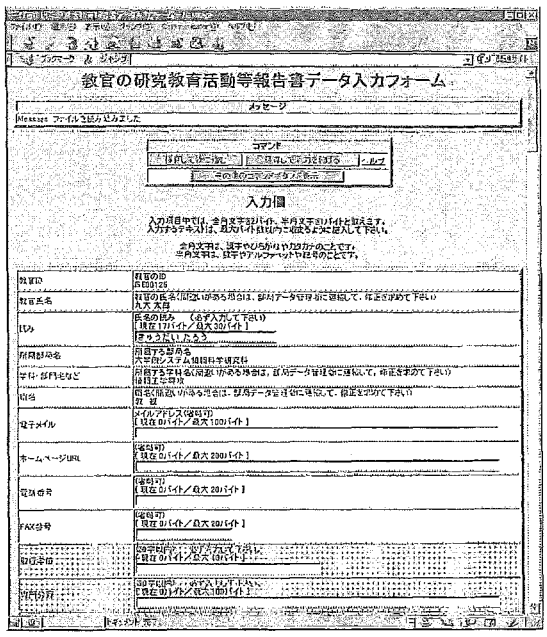


図 2　報告書の入力画面
Fig. 2　Snapshot of inputting a report.

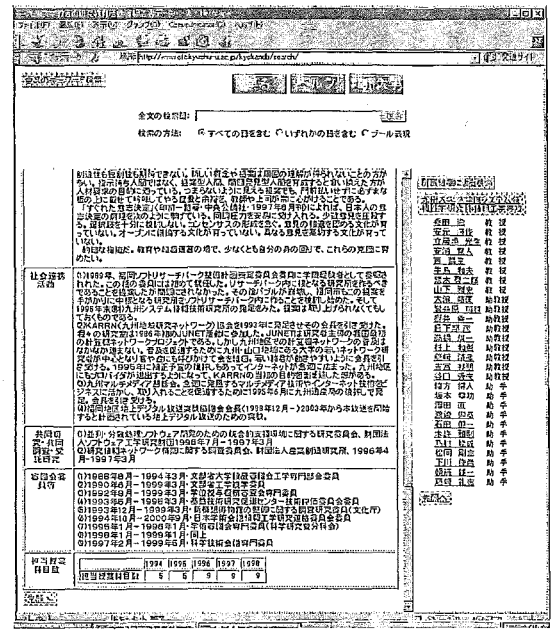


図 3　公開データの表示画面
Fig. 3　Snapshot of displaying a data opened.

損失の被害が大きいと考えられる．報告書の入力を一度の送信で行うようにすると，送信が失敗したときにすべてのデータが消えてしまう．この問題に対応するために，提出前のデータをサーバ側の中間データに一時的に保存しながら少しずつ入力ができるようにした．

図 2 は，直接入力の画面である．

b）　部局データ管理者による代行入力

ブラウザの操作が行えない状況にある教官に代わって，部局データ管理者が，書面により提出された報告書を入力することができる．

c）　教官によるテンプレート入力

テンプレートとは，報告書の SGML [5] フォーマットに基づくひな形である．入力を必要とする項目名や制約は SGML によるコメントとして参照され，各入力項目は各タグに対応している．したがって，各タグ内に各項目の内容を入力することで，教官がテキストエディタなどを用いて報告書の入力を行うことができる．

d）　部局データ管理者によるテンプレート代行入力

部局データ管理者は，依頼を受けた教官のテンプレートを取得して，教官に電子メールなどの手段で送信することができる．また，テンプレートを利用して作成された報告書が，教官から電子メールなどで送られてきた場合にも，それを検査した後，サーバに送信して公開することが可能である．この方法により，電子メールによる報告書提出に対応することができる．また，セキュリティ上の都合で教官の入力フォームに対して組織外からのアクセス制限をかけた場合，学外に出張中の教官は入力を行うことができなくなってしまうような問題にも対処できる．

3.1.3　データベースの公開

各教官が入力を終えて，提出した後に部局データ管理者の検査を経て公開されたデータは，閲覧者が自由に閲覧することができる．報告書を検索する方法は，報告書のキーワード入力による検索，組織に基づいた検索の二通りがある．

報告書のキーワード入力による全文検索は，閲覧者がキーワードを報告書検索システムに入力することによって，そのキーワードを含む報告書を提出した教官の一覧を閲覧者に提示する．

組織に基づいた検索を用いるときは，教官の所属する組織を選択することによって，教官の報告書データ一覧を得ることができる．

公開データは静的な HTML ファイルとして公開さ

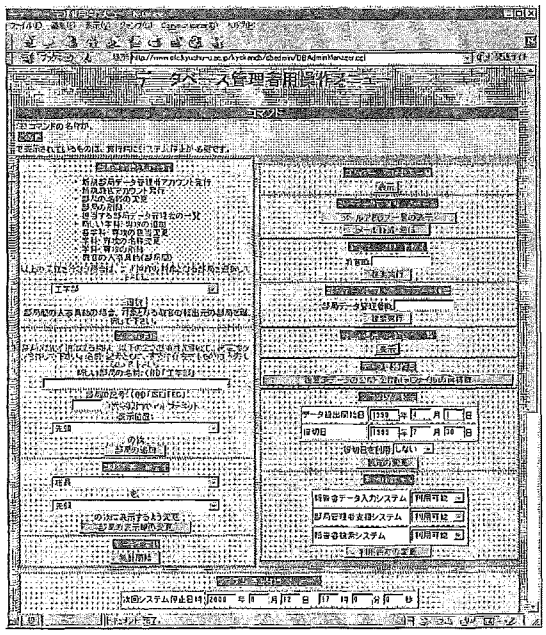


図 4 データベース管理者のユーザインタフェース
Fig. 4 The user interface for the database administrator.

れているが，データベースに変更があったときは対応する HTML ファイルを再構築するようになっている．

図 3 は，公開データを表示している画面である．

3.1.4 データベースの保守

本システムのデータベースにおいては，報告書の入力や公開以外にも，新規アカウントの発行などのシステムの運用が必要になる．しかし，九州大学のような数千人規模の教官を対象とした運用の場合，保守作業が発生する回数は 1 人の管理者では処理できないものになる．したがって，本システムは，各部局ごとの分散管理が行えるシステムとなっている．管理者に提供する機能は以下のとおりである．

a) データベース管理者のための機能

各モジュールの利用許可及び禁止の設定を行うサービス制御，部局データ管理者アカウントの管理，部局の新設や削除など，大学全体のシステム管理に伴うデータベース操作の機能がある．図 4 は，データベース管理者のためのユーザインタフェースである．

b) 部局データ管理者のための機能

管理している教官の一覧表示，提出された報告書をプレビュー表示して公開若しくは返却の決定，教官アカウントの管理といった機能がある．特徴的な機能として，部局にまたがる教官の移動がある．移動元と移動先が別々の部局データ管理者の担当である場合，まず，移動元の部局データ管理者が移動先の部局データ管理者に対して移動要求を送る．移動先の部局データ管理者は，受理するか否かを決定する．受理されれば，移動が成立する．この手順はデータベース管理者の手間を必要としない．また，削除された教官のアカウントはバックアップとして保存されて，部局データ管理者が参照することができる．

3.2 データベース設計

報告書データは，リレーショナルデータベース管理システム（以下，RDBMS）の管理下に置かず，SGML テキストとして施錠及びバックアップ機能を付加したファイルシステム内に格納する．各 SGML データのフォーマットは，リレーショナルデータベース内に格納されているメタデータによって規定される．メタデータには，各要素についての型情報や，文字長の制約，初期値といった情報が含まれている．1 種類のメタデータは一つのテーブルに格納されており，各報告書 SGML データには，メタデータとしてどのテーブルを利用するのか，という情報が含まれている．したがって，異なったフォーマットをもつ報告書を混在させることができる．すなわち，部局や年度ごとに異なったフォーマットを利用できる，ということである．

以下に，SGML データベースのテーブル一覧を示す．

kyokan　各教官の公開された報告書．

kyokan_submit　各教官の，提出されたが未検査の報告書．

kyokan_tmp[a-z]　各教官の報告書の中間データ．

残りのテーブルは，すべてリレーショナルデータベース内のテーブルである．そのうち，報告書に関するテーブル，並びに **3.1.4** に示した保守機能に関するテーブルの一覧を以下に示す．

changeplan　部局間移動の要求．教官 ID を主キーとして，移動元，移動先の組織．

kyokan_nyuryokubi　各報告書の入力日．SGML データベースに含めてしまうと，整列のために行わなければならないファイル検索の手間が大きいので，リレーショナルデータベースに含まれている．

kyokan_torokubi　報告書の登録日．同上．

kyokanidalias　教官 ID の別名．移動によって ID が変わっても旧 ID でアクセスできるようにするため．

oldsoshiki　削除された教官のアカウント．

soshiki　教官のアカウント．教官 ID を主キーと

する．

sys_*　各テーブルのメタデータ．

各教官に対して複数の中間データを管理することにより，SGMLデータがもつデータに冗長な項目が発生した．同一の教官のデータであれば同じ値をもつ，教官の職名や，報告書の提出日である．このようなデータの更新時にすべてのデータを更新しなければならない手間を考えると，SGMLにこれらのデータを含ませるのは避けたかった．また，これらの項目は，報告書データのフォーマットが変更されても，常に不変であると特定されたため，リレーショナルデータベースに含まれるsoshiki，kyokan_torokubiテーブルを参照する．

3.3　システムアーキテクチャ

システムを構成するサーバは，データベースを管理するRDBMSとそれにアクセスするためのユーザインタフェースを提供するCGI（Common Gateway Interface）プログラムからなっている．プログラミング言語としては，Perl5 [7]を使用している．RDBMSにはpostgreSQLを用いた．

本システムは，七つのサブシステムから構成されている．サブシステムへの分割に関して留意した点がいくつかある．まず，データベース管理システムの変更によって生じるソフトウェアの変更をなるべく少なくするためにデータベースへのアクセス部をサブシステム化して，すべてのデータベースへのアクセスはデータベースアクセス部を通じて行うようにした．また，データベース管理者，部局データ管理者，教官のそれぞれに適したユーザインタフェースをサブシステム化した．また，メタデータによって教官ごとに異なる報告書のフォーマットを用いることができるということは，開発の初期段階では必要のない機能であったが，将来必要とされる可能性の高い機能であった．したがって，そのような機能を利用する可能性のある，すなわち，変更が予測された部分である，入力フォーム作成部やデータ公開部をサブシステム化してそれに備えた．

本システムのモジュール構成を，図5に示す．

3.3.1　データベースアクセス部

データベースを管理するRDBMSとのインタフェースとなるサブシステムである．本サブシステムは，以下のモジュールで構成されている．

DBAdapterモジュールは，RDBMSやSGMLデータベースに対する様々な要求のインタフェースをもつ．

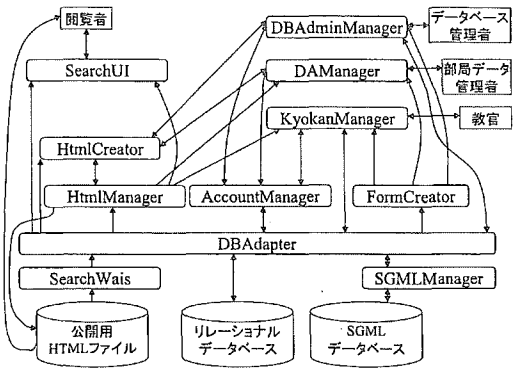


図5　システムアーキテクチャ
Fig. 5　The system architecture.

RDBMSにアクセスする場合，このモジュールが上位層からの要求に対して対応するSQLを発行する．

SGMLManagerモジュールは，SGMLで管理されているテーブルへのアクセスのためのインタフェースをもつ．

3.3.2　アカウント管理部

本サブシステムに含まれるAccountManagerモジュールは，教官及び部局データ管理者のアカウント管理に関するインタフェースをもつ．

3.3.3　教官データ入力支援部

本サブシステムに含まれるKyokanManagerモジュールは，各教官と本システムとのインタフェースとなる部分である．各教官は，ネットワークを通じて本サブシステムのCGIプログラムに対して要求を行う．

3.3.4　入力フォーム作成部

本サブシステムに含まれるFormCreatorモジュールは，報告書の入力フォームをHTMLフォーマットで作成する機能をもつ．

3.3.5　データベース管理者支援部

本サブシステムに含まれるDBAdminManagerモジュールは，データベース管理者のためのユーザインタフェースである．

3.3.6　部局データ管理者支援部

本サブシステムに含まれるDAManagerモジュールは，部局データ管理者のためのユーザインタフェースである．

3.3.7　データ公開部

検査後の報告書を公開するためのサブシステムである．

HtmlManagerモジュールは，報告書データを，Web

ページとして公開するための HTML フォーマットに変換するためのモジュールである．また，リレーショナルデータベースに含まれる組織構成に関するテーブルから，組織による報告書検索を可能にするための HTML データを生成する機能をもつ．

HtmlCreator モジュールは，他のモジュールからのデータベース変更通知に従って，公開 HTML ファイルのうち更新が必要な部分を再構築する．HTML データの生成には HtmlManager を利用する．

SearchUI は，公開データの全文検索を可能にするためのユーザインタフェースを提供する．

SearchWais は，Wais を用いて公開データの全文検索を行い，結果を処理する．

4. 運用とシステム更新

本章では，本システムの運用と，運用中にシステムを更新するために用いたリリースフローについて述べる．

4.1 運用体制

本システムの開発から現在に至る期間は，次の四つの期間に分かれた．

システム開発期間（～1997/12/02）：システムの仕様設計及び開発を行った．

システム運用テスト期間（1997/12/03～1998/04/15）：一部の部局に報告書データを入力してもらい，システムの運用テストを行った．

データ入力期間（1998/04/16～1998/09/21）：全学において報告書データを入力した．

公開運用期間（1998/09/22～）：入力された報告書データを一般に公開した．

また，全期間において，利用者や開発者から得られた報告や要望をもとにバグ修正及びシステムの改善を行った．

4.2 リリースプロセス

システム運用テスト期間以降は，本システムを更新する際の手順を策定し，バグ修正やシステムの改善の際に適用した．この手順をリリースプロセスと呼ぶ．リリースプロセスの概要を図 6 に示す．更新したシステムの動作確認をするためにテスト用システムや，各開発者が使用する作業用領域を用意し，バージョン管理システムを用いて本システムのバージョン管理と一貫性管理を行った．

また開発者のうち 1 名が，リリースプロセスを管理するシステム更新管理者を担当した．

以下では，リリースプロセスの詳細を示す．

［利用者からの質問または要望］ 利用者が本システムに関して質問あるいは要望がある場合は，まずデータベース管理者にその内容を報告する．システムの使用方法についての質問などデータベース管理者が自分で対応できるものに対しては，データベース管理者が利用者に対し適切な返答をし，このプロセスを終了する．バグ修正要求やシステムの改善要求といった，開発者の対応を必要とする内容であれば，開発者に報告し，方針決定作業に進む．

［開発者からの要求］ 開発者からバグ修正要求やシステムの改善要求が発生した場合は，直ちに方針決定作業に進む．

［方針決定作業］ この作業では，開発者同士で以下の内容を決定する．

- 要求への対処が，単なる返答で十分なのか，バ

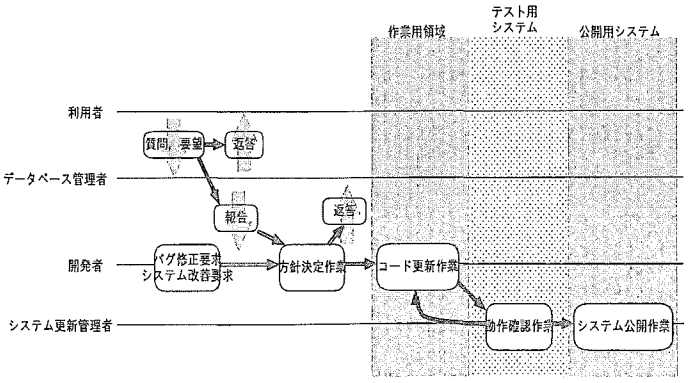


図 6 リリースプロセス
Fig. 6 The release process.

グの修正が必要なのか，あるいはシステムの改善が必要なのかを決定する．単なる返答で十分な場合，開発者は利用者またはデータベース管理者に適切な返答をして終了する．このときのリリースプロセスを質問プロセスと呼ぶ．

- バグ修正あるいはシステムの改善が必要な場合は，どのモジュールを変更すればよいのかを特定し，担当する開発者を決定する．バグ修正が必要な場合のリリースプロセスをバグ修正プロセスと呼び，システムの改善が必要な場合のリリースプロセスをシステム改善プロセスと呼ぶ．

[コード変更作業] 方針決定作業で割り当てられた開発者は，個人の作業用領域でコードの変更を行う．変更を終えると，テスト用システムに変更を反映し，システム更新管理者に報告する．

[動作確認作業] システム更新管理者は，コード変更作業にかかわったすべての開発者から作業終了の報告を受けとった後，テスト用システムにおいてシステムの動作確認を行う．この段階で問題が見つかれば，システム更新管理者は開発者に報告し，コード変更作業へ戻る．

[システム公開作業] システム更新管理者は，変更を本システムに反映する．その際，本システムのすべてのサービスを停止する必要がある．しかし本システムは HTTP による状態なしの通信を行うため，停止する際に使用している利用者を把握することが難しく，教官がデータを入力している途中にサービスを停止してしまう危険がある．そこで，サービスを停止するのは平日の夜や休日とし，事前にサービスを停止する旨のメッセージをシステムに表示することとした．この方法を採用したのは，本システムが日本語で公開されるページであり，利用者が少ない時間帯であることが予測できたためである．変更を本システムに反映した後，システム更新管理者は変更の影響を受けるドキュメントを修正し，このプロセスを終了する．

リリースプロセスが開始して終了するまでの時間を，リリースプロセスの生存時間と呼ぶ．

5. システム及び運用体制の評価

本章では，本システムで用いた Web サーバのログ，及び本システムのログ，やり取りされた電子メールのログ，バージョン管理システムのログを解析し，システム及び運用体制を評価した結果を示す．

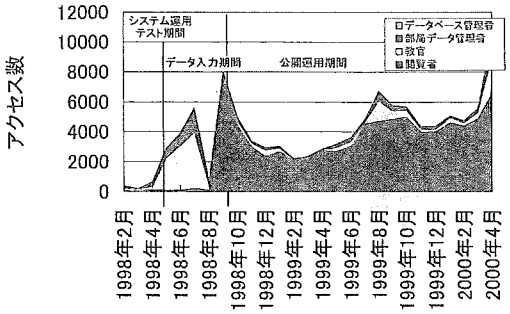


図 7 本システムへのアクセス数
Fig. 7 The number of accesses to the system.

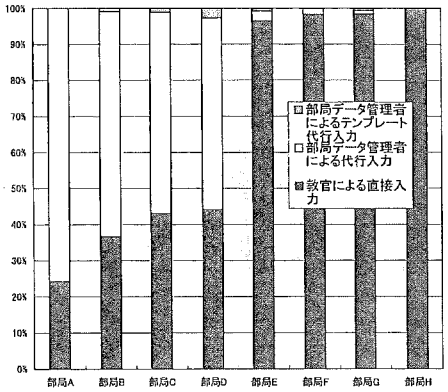


図 8 部局ごとのデータベース入力数
Fig. 8 The number of inputs at each department.

5.1 システム運用結果

図 7 は，Web サーバのログを解析して得た，本システムへのアクセス数を示すグラフである．このグラフでは，アクセス数を，閲覧者用，教官用，部局データ管理者用，データベース管理者用のページに分けて示している．

図 8 は，本システムのログを解析して得た，データベースへの入力の利用状況を示すグラフである．このグラフでは，報告書の入力数を，教官による入力，部局データ管理者による代行入力，部局データ管理者によるテンプレート代行入力に分けて部局ごとに示している．ただし教官による入力とは，教官による直接入力と教官によるテンプレート入力の和である．教官による入力の内訳は，教官によるテンプレート入力機能が教官による直接入力機能の一部として実装されたため，ログを得ることができなかった．また，部局は，教官による入力の比率が最も低い部局と最も高い部局

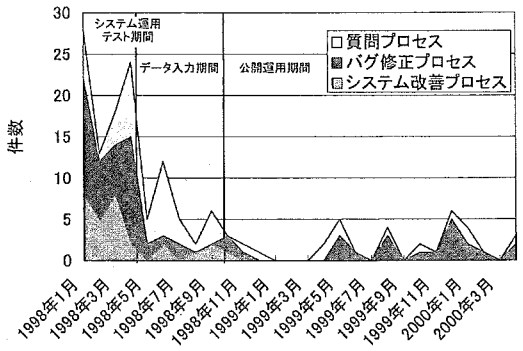


図 9 リリースプロセス発生回数の推移
Fig. 9 Passage of the number of release processes.

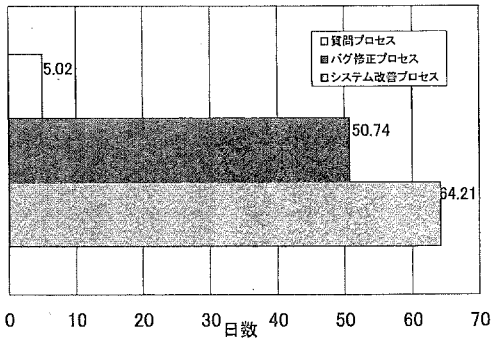


図 10 リリースプロセスにおける平均生存時間
Fig. 10 The average lifetime in release processes.

から 4 部局ずつとっている.

本論文の以降の解析結果は，電子メール及びバージョン管理システムのログの解析から得られたものである.

図 9 は，バージョン管理システム導入以降においてリリースプロセスが発生した回数の変化を示すグラフである．内訳として，質問プロセス，バグ修正プロセス，システム改善プロセスを示している.

図 10 は，バージョン管理システム導入以降におけるリリースプロセスの生存時間を，リリースプロセスの種類ごとに平均をとったグラフである.

5.2 システム更新の記録

5.2.1 モジュールの更新

図 11 は，バージョン管理システム導入以降における各モジュールの更新回数を示すグラフである.

図 12 は，バージョン管理システム導入以降において，モジュールの更新を含むリリースプロセスが更新したモジュールの個数の分布を示したグラフである.

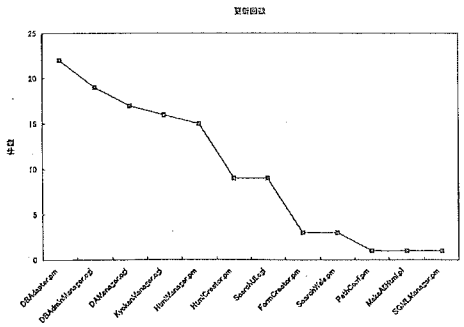


図 11 各モジュールの更新状況
Fig. 11 The status of updates at each module.

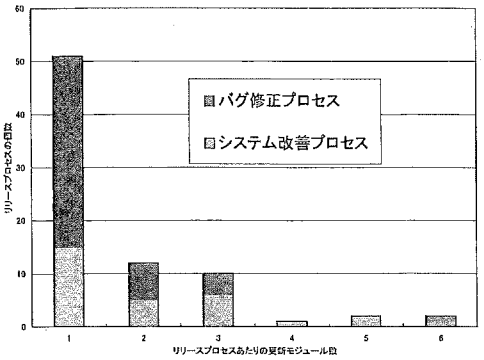


図 12 リリースプロセスがモジュールを更新した個数の分布
Fig. 12 The distribution of the number of updated modules in release processes.

5.2.2 データベーススキーマの変更

リリースプロセスの中で起きたデータベースの更新には，次のようなものがあった.

a) 部局をまたがる教官の移動に必要なテーブルの追加

3.1.4 b) に示したような，異なる部局間の教官移動を部局データ管理者が行う機能の要求が，システム運用中に発生した．そのため，移動先の部局データ管理者の処理を待つ教官の情報を格納するテーブル changeplan 及び，移動した後の教官が移動前の情報を参照するためのテーブル kyokanidalias を追加した.

b) 報告書の中間データのための SGML ファイルの追加

3.2 で示したような中間データの保存のため，kyokan_tmpa から kyokan_tmpz までの SGML ファイルを追加した.

c) 各教官が複数の報告書データをもつことへの対応

3.2 のとおり，重複し，かつ教官について一意に定まる項目をリレーショナルデータベース soshiki と kyokan_torokubi に移動して解決した．

d) データベースアクセスの高速化のための変更

SGMLデータのうち頻繁にアクセスされ，かつ複数のSGMLファイルに分散していた報告書の入力日と公開日の項目を，新しいリレーショナルデータベースのテーブル kyokan_nyuryokubi 及び kyokan_torokubi に移動して高速化した．

5.3 考　　察

5.3.1 システムに関する考察

図7より，公開運用期間中は，閲覧者のページへ月2000件以上，最高8000件近くのアクセスがあることがわかる．また，教官のページへのアクセスは，データ入力期間中は最も多いものの，公開運用期間になると減少している．ただし常にアクセスはされており，本システムにより教官が随時報告書を更新できていることが推測できる．データベース管理者のページへのアクセス数は非常に少なく，このグラフからは確認することができない．部局データ管理者を置くことにより，データベース管理者の負担が削減されたと考えられる．

図8より，いくつかの部局においては部局データ管理者による代行入力が行われていることがわかる．このことから，**3.1.2** b) により部局データ管理者機能の中から教官のデータを代行入力できるようにしたことの有効性が確認できる．また，**3.1.2** d) に述べた部局データ管理者によるテンプレート代行入力は，割合は少ないながらも行われている．この理由としては，海外出張などで組織外から入力が必要な教官にとって部局データ管理者によるテンプレート代行入力が必要であったことが推測できる．

図11より，最も更新が多かったのは，データベースに直接アクセスする DBAdapter であり，次にデータベース管理者のための DBAdminManager，部局データ管理者のための DAManager，教官のための KyokanManager，データ公開部に関するモジュールと続く．DBAdapter の更新頻度が高かったのは，DBAdapter がすべてのデータベースアクセスを担当しており，データベースへのアクセスに関する変更の際には DBAdapter の修正が必要であったからである．また，DAManager，KyokanManager，DBAdminManager の更新頻度が高かったのは，本システムの中で複雑な作業を要求されるのが教官，部局データ管理者，及びデータベース管理者であるため，それらの利用者への利用者インタフェースを幾度か試行することにより改良する必要があったためである．

また，図12より，多くのリリースプロセスが一つのモジュールの修正ですんだことがわかる．特にバグ修正プロセスの場合に顕著であり，システム改善プロセスの場合も更新モジュール数が少ないほどリリースプロセスの個数が減少する傾向にある．このことから，システム更新時の修正箇所が複数のモジュールにわたることが少ないことが確認でき，モジュール分割が適切であったといえる．

5.2.2 a) は，複数の利用者の協調による処理が必要な機能を追加する必要がある場合に，新たなデータベースを追加して情報をやり取りする1例を示している．**5.2.2** b) c) は，教官という一つの属性に対し管理すべきデータの多重度が増え，かつ増加したデータがSGMLファイルで管理されていた場合に，SGMLファイル中にあるが多重度が増えないデータを一つのテーブルにまとめることで一貫性の確保を図った例である．**5.2.2** d) は，SGMLデータをファイルのまま管理したことにより生じた処理速度の問題を，RDBMSに一部データを重複させることで解決した例である．この効果を測定するため，移動した項目を頻繁に使う機能である教官の一覧表示を実行するための CGI プログラムの応答時間を，SGMLデータを検索する場合と，RDBMSに格納されたデータを検索する場合において比較した．117人の教官をもつ部局に対する試行を10回行ったところ，前者は平均328.8秒，後者は29.8秒という結果だった．このことから，この変更の効果を確認することができる．この方法は重複したデータをもつため検索に対して効果があるが，データの更新は逆に時間がかかる．別の解決策として，SGMLをリレーショナルデータベースに変換して格納する手法が考えられる．

5.3.2 運用体制についての考察

図9より，リリースプロセスの発生回数はシステム運用テスト期間からデータ入力期間に移行する際に減少している．データ入力期間では全学の教官が入力するため利用者が大幅に増えたことを考えると，システム運用テスト期間に行われたテストが効果を上げていることがわかる．また，データ入力期間中にリリースプロセスは全体として減少する傾向にある．これは，

データ入力期間が公開運用期間に向けてのテスト期間の役割を果たしたと考えることができる．

図 10 より，質問プロセスの生存時間は極端に短く，システム改善プロセスの生存時間が最も長い．このことと，図 9 でバグ修正プロセス，システム改善プロセスの和がシステム運用テスト期間で最も大きくなっていることを考えると，開発者の負担はシステム運用テスト期間に最も大きくなっていることがわかる．システム運用テスト期間と公開運用期間ではどちらが開発者の負担が大きいかは明らかではないが，最も平均生存時間が長いシステム改善プロセスがシステム運用テスト期間に多く含まれることを考えると，システム運用テスト期間の方が負担が大きいことが推測できる．

6. 関連研究

6.1 ソフトウェア開発プロセスの分析について

本論文では，閲覧者，教官，部局データ管理者，データベース管理者，(これらは，広い意味でユーザと呼んでもよい) と，開発者の関係に注目して，ソフトウェアの開発プロセスについての分析を行った．これによって，運用開始後における開発者の役割と，負担の量についての事例を示すことを一つの目標としている．

ソフトウェアの開発プロセスにおいて，詳細なバグの履歴をとり，それらについて，内省とでもいえる解説を加えた論文として，Knuth の [2] が挙げられる．この論文は，Knuth がほぼ 1 人で開発した TEX という組版システムにおける誤りについての詳細なレポートである．我々の論文では，この研究を範としながら，複数人の開発者によって，誤りの履歴をとっている．また，ユーザと開発者のやり取りの中で仕様が変更されていった過程を分析することによって，ユーザサポートを開発者が行うことによる負担とその経年変化について数値化を行っている．これらの作業には，バージョン管理システム CVS が大きく貢献している．また，ユーザと開発者，あるいは開発者間における連絡や議論のほとんどを電子メールによって行ったということも後々に開発プロセスを分析する際の大きな手助けとなっている．

また，ソフトウェア信頼性を数理モデルによって定量的に推定・予測する研究が行われている [8]．つまり，実際の開発において発生する誤りの数をあらかじめ予測することによって，実際の開発過程におけるある段階の信頼性を見積もろうという考え方である．

これらの研究が誤りの発生数，つまり量に焦点を当てているのに対して，本論文では，開発者に報告された誤りの数のみならず，その内容，いわば，質についても着目している．例えば，誤りや仕様変更の種類について記録し，それらを行う作業について，どれぐらいの時間をかけて処理されたかを追跡している．これらは，開発者とユーザとの電子メールのやり取りをすべて解析することによって行っている．

ソフトウェア開発後に，開発者が運用にかかわる必要があることが多くの場合必要とされる．本論文は，それが具体的にどのような負担となっているのかを示す具体例を含んでいる．

6.2 大学における教官の研究教育活動公開システムについて

大学における教官の研究教育活動をインターネットを用いて公開するという同じ目的をもつシステムがいくつか公開されている．

九州共立大学工学部教員情報入力・発行システム [1] は，大学における教官紹介やシラバスのような統一仕様の Web ページを，教官がブラウザの入力欄を用いて入力し，インターネットに公開することができるというシステムである．教官紹介という目的や，各教官がオンラインで入力するという点が本論文のシステムと類似している点である．本論文のシステムにない特徴は，印刷のために PDF (Portable Document Format) 形式で出力する機能が備わっている，ということがある．このシステムになく本論文のシステムにある特徴は，公開されるデータの管理者による検査が行えること，組織の変更などに関する管理機能の充実，また，全文検索機能があることを挙げることができる．

東京大学研究者紹介 [6] は，本論文のシステムと同様に，一般社会に研究者情報を公表することを目的として開発されたシステムである．大学の組織だけではなく，研究分野の選択によって検索する方法が用意されている点が本論文のシステムにない特徴である．ただし，紹介の内容には，研究者の自由作文による研究紹介はない．

神戸大学研究者紹介 [3] も同様の目的をもつシステムであるが，項目を限定した全文検索が行える点が異なっている．逆に，全項目を対象とした全文検索は行えない．

7. むすび

本論文では，九州大学教官の研究教育活動等報告書データベースについて，そのシステム構成，運用体制

という二つの視点から論じた．利用者の要求や開発者の議論によって行われた仕様の変更についての履歴や，九州大学で実際に過去2年半に渡って運用したことで得られたシステムのログや開発者グループに送られた電子メールを分析することによって，システム構成，運用とシステム更新の体制についての評価を行った．それにより，データベース管理者の下に複数の部局データ管理者を置いて分散管理を行う，適切なモジュール分割を行いシステム更新時の修正箇所を集中させる，教官が自ら入力することによって更新頻度を上げるといった，開発当初の目標が達成されていることが確認できた．設計段階で十分な検討を重ねたこと，運用規模を考慮して管理負担の増大を事前に予測できたことがこれらの成功に繋がったといえる．

2000年度より，九州大学では，研究と教育における柔軟な協力・交流・移動を実現することを制度的に確立するための組織改革による新組織が導入された．本システムについても，この改組に伴う大幅なソフトウェアの更新が必要とされている．今後は，この新組織導入に伴うソフトウェア更新について検証・評価し，当初の設計から考慮していた組織の変更に対する柔軟性についての考察を行わなければならない．

文　献

[1] 安在弘幸，宿輪憲司，平原貴行，“統一フォーマットをもつホームページの分散入力・自動発行システム，” 第13回情報処理学会九州支部研究会報告，pp.278–285, March 1999.

[2] D.E. Knuth, “The errors of TeX,” Software — Practice and Experience, vol.19, no.7, pp.607–685, 1989.

[3] “神戸大学研究者紹介，” http://ref.ofc.kobe-u.ac.jp/

[4] “九州大学教官の研究教育活動等報告書データベース，” http://www.ofc.kyushu-u.ac.jp/kyokandb/

[5] International Organization for Standardization, “Information processing — Text and office systems — Standard Generalized Markup Language (SGML),” ISO 8879:1986, 1986.

[6] “東京大学研究者紹介，” http://www.adm.u-tokyo.ac.jp/researcher/index.htm

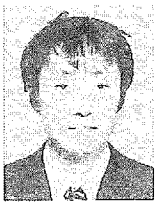
[7] L. Wall, T. Christiansen, and J. Orwant, Programming Perl, 3rd Edition, O'Reilly, Cambridge, 2000.

[8] 山田　茂，“ソフトウェアの品質評価に関する考え方と動向—ソフトウェア信頼度成長モデルに基づく定量的品質評価法，” 情報処理，vol.32, no.11, pp.1189–1202, 1991.

（平成12年8月25日受付，12月22日再受付）

野中　裕介

1999九大大学院システム情報科学研究科情報工学専攻修士課程了．同年4月より同専攻博士後期課程に在籍．ソフトウェア工学，並行システム開発手法に興味をもつ．情報処理学会，IEEE-CS各会員．

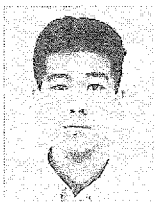


井上　創造

1997九大・工・情報卒．1999同大大学院システム情報科学研究科情報工学専攻修士課程了．同年4月より同専攻博士後期課程に在籍．データベースの協調作業支援への応用の研究に従事．情報処理学会会員．

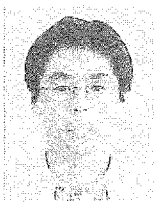
秦野　克彦

1999九大・工・情報卒．2001同大大学院システム情報科学研究科情報工学専攻修士課程了．ソフトウェア工学及びソフトウェア開発支援環境に興味をもつ．



原田　　努

1999九大・工・情報卒．2001同大大学院システム情報科学研究科情報工学専攻修士課程了．ソフトウェア工学及びネットワークセキュリティに興味をもつ．

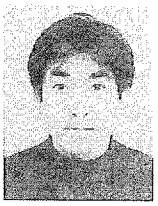


乃村　能成

1993九大・工・電子卒．1995同大大学院情報工学専攻修士課程了，同年九州大学工学部助手，1996九州大学大学院システム情報科学研究科助手．ソフトウェア開発環境・スライシング技術に興味をもつ．情報処理学会会員．

岩井原瑞穂　（正員）

1988九大・工・情報卒．1991年1月から1年間，カナダ・マッギル大学大学院工学研究科計算機科学専攻に留学．1993九州大学大学院工学研究科情報工学専攻博士後期課程了．博士（工学）．同年4月より同大学院総合理工学研究科助手．1996年5月同大学院システム情報科学研究科助教授．1999年9月より1年間米国ジョージア大学計算機科学科客員研究員．2000年4月九州大学大学院システム情報科学研究院助教授，現在に至る．データベース質問処理，質問言語，協調作業支援等の研究開発に従事．情報処理学会，ACM，IEEE各会員．



峯　恒憲

1987 九大・工・情報卒．1992 同大大学院総合理工学研究科情報システム学専攻博士課程単位取得退学．同年，同大教養部情報科学教室講師．1994 同大理学部講師．1996 同大大学院システム情報科学研究科（現・研究院）助教授，現在に至る．1998 年 9 月から 1999 年 4 月までドイツ人工知能研究所（DFKI）客員研究員．1999 年 4 月から 1999 年 9 月までカーネギーメロン大学言語技術研究所（LTI）客員研究員．人間の認知能力に興味をもち，自然言語処理，情報検索，情報抽出，エージェントシステム等の研究開発に従事．情報処理学会平 4 年度論文賞受賞．博士（工学）．情報処理学会，人工知能学会，言語処理学会，ACM，AACE 各会員．

牛島　和夫　（正員）

1961 東大・工・応用物理（数理工学コース）卒．1963 同大大学院修士課程了．同年九州大学中央計数施設勤務．1977 九州大学工学部情報工学科教授（計算機ソフトウェア講座担当），1996 同大大学院システム情報科学研究科（現・研究院）教授，現在に至る．1990 年 4 月から 1994 年 3 月まで九州大学大型計算機センター長，1996 年から九州大学大学院システム情報科学研究科長（現・研究院長）を兼務．1987 年 5 月から 1989 年 5 月まで情報処理学会理事，1988 年 5 月から 1990 年 5 月まで本会評議員．1991 年 5 月から 1992 年 5 月まで本会・情報・システム研究グループ委員長．1992 年 5 月から 1993 年 5 月まで本会九州支部長．1992 年 5 月から 1994 年 5 月まで本会評議員．1995 年 5 月から 1997 年 5 月まで情報処理学会監事．1999 年 5 月から 2000 年 12 まで IEEE Fukuoka Section Chair．工博．情報処理学会，ソフトウェア科学会，ACM，IEEE-CS 各会員．