インターネット利用者を対象とした
情報収集機構に関する研究

東京工業大学 大学院
情報理工学研究科 数理・計算科学専攻
酒井 淳一
(学籍番号 97M37147)

平成10年度修士論文

指導教官 大野 浩之 講師

平成11年1月29日
(平成11年2月24日改訂)

目 次

第1章　はじめに

人々が問題意識を持ったとき、何らかの手段で結論を導き、問題を解決していく。本研究は、情報収集を迅速化することで、迅速に結論を導くことを目的とする。特に人々を対象として情報収集することを考える。人々を対象とした情報収集は、意志決定や統計調査などに有効であり、情報収集手段として選挙やアンケート調査がよく利用される。しかし、これらの手段には多くの人手と時間がかかる。

ところで、現在多くの人々がインターネットを利用して情報収集している。情報収集の対象や目的に応じて、WWW 検索エンジン、archie などが利用されている。また、人々を対象とした情報収集も多くなされている。例えば、インターネット上で学会の受け付けや、統計調査、マーケティング調査などがなされている。インターネットは大量の情報を迅速に得るのに適している。

そこで、インターネットを利用して、迅速に情報収集することを考えた。しかし、インターネット利用者を対象とした情報収集には、多くの要件を満たす必要がある。例えば、得られるデータの信頼性の下げないために、通信内容の機密性、完全性を保持する必要がある。また、回答者のプライバシを保護しつつ、回答者を認証し、多重回答を防止する必要がある。現在では、これらの要件を実現するための情報収集プロトコルが提供されておらず、要件のすべてが解決されているとはいえない。情報収集者は WWW や電子メールなどの既存の情報交換プロトコルを利用するしかない。

これを受けて筆者は、rics と呼ばれる情報収集システムを開発した。rics は、階層構造を持ち、下位層から rics プロトコル、rics データフローモデル、rics アプリケーションと呼ぶ。rics プロトコルは通信方式を規定し、rics データフローモデルは rics アプリケーション間のデータの流れを規定する。rics プロトコルおよび rics データフローモデルは既存の情報交換システムに依存しない形で提供され、回答者のプライバシを保護し、信頼性の高いデータを得ることを保証する。rics アプリケーションは rics プロトコルで通信し、rics データフローモデルに従いデータを交換する。rics アプリケーション間の通信により情報収集がなされ、簡単かつ迅速な情報収集を実現する。

本論文では、rics の各階層が果たす役割、用いられている仕組み、要素技術を中心に述べる。そして、rics による情報収集の安全性について議論する。さらに、rics をいくつかの場面で実験、運用し、さまざまな情報収集に有効に機能することを示す。

第 2 章では、研究の背景および目的を述べ、インターネット利用者を対象とした情報収集に必要な条件について検討する。第 3 章では、要件を満たすための研究、技術、仕組みを検討する。第 4 章では、要件を実現するシステムの設計と実装について述べる。第 5 章では、rics の安全性について評価および考察する。第 6 章では、rics の利便性について評価および考察する。第 7 章で今後の展望を述べ、第 8 章でまとめを述べる。

第2章　研究の背景と目的

この章では、まず研究の目的を述べる。次に、インターネットを利用することの有用性を述べる。そして、インターネット利用者を対象とした情報収集を定義、分類し、その必要条件を検討する。

2.1　研究の目的

問題意識を感じたとき、人々は何からの手段で結論を導き、問題を解決する。結論を導く手段が情報収集である場合、その流れを以下のように考える。

1. 問題意識を持つ
2. 問題を明確にする
3. 情報収集する
4. 収集した情報を解析する
5. 結論を導く

本研究は、結論を導くまでの段階のうち、情報収集の段階に焦点を当てる。情報収集の目的はさまざまであるが、情報収集の段階を迅速化することにより迅速に結論を導くことを目的とする。情報収集の対象としてさまざまものが考えられるが、特に人を対象とした情報収集に焦点を当てる。

情報収集を迅速化するだけでは、迅速な問題の解決にはつながらない。誤った結論を導く可能性があるからである。ゆえに、信頼性の高い情報を迅速に得ることが必要である。また、情報収集の目的はさまざまであるため、その規模もさまざまである。情報収集機構はスケーラビリティを持つ必要がある。

2.2　インターネットの利用

人々を対象とした情報収集には、アンケート調査や選挙などがよく利用される。しかし、これらの手段は多くの人手と時間がかかる。ところで、現在多くの人々がインターネットを利用して情報を収集している。情報収集の対象や目的に応じて、WWW の検索エンジン、archie、whois などが情報収集手段として利用される。インターネットを利用すれば、さまざまな情報を大量そして迅速に収集できる。筆者は、インターネットを利用することが情報収集の迅速化に有効であると考えた。インターネット利用者を情報収集の対象とした場合、以下のような利点があると考える。

- 費用が少ない。
- 人手が少ない。
- 回答データが電子化されているので、集計の手間がかからない。
- 迅速に回答データを回収できる。

今日、インターネット上でも、その利用者を対象とした情報収集が多くなされている。日本でもインターネット利用者は 1000 万人を越え [1]、情報収集の対象として無視できなくなった。学会やセミナーの受付では、インターネットを利用して、主催者が参加者から参加者の個人情報を取得する。インターネット上の通信販売では、店舗が商品および購入者の個人情報を購入者から取得する。また、インターネットを利用した統計調査も実施されている。

インターネット利用者からの情報収集はビジネスとしても成り立ちつつある。インターネット上でマーケティング調査や統計調査を代行する組織には、iMi ネット[1] 、HI-HO マーケティングサービス[2] 、KNOT'sClub[3] 、JapanView[4] などがある。どの組織も会員を持ち、その会員に対してアンケート調査を実施している。iMi ネットは電子メール、HI-HO マーケティングサービス、KNOT'sClub、Japan View は WWW を利用している。インターネットを利用して授業調査アンケートも実施されている [2]。また、企業も人々からの情報収集に積極的になりつつある [1]。

本研究では、人々を対象とした情報収集手段としてインターネットを利用する。しかし、インターネット利用者を対象とした情報収集には求められる技術および条件が多くあり、その全てが解決されているとはいえない。要件をすべて実現させるためには、情報収集者の手間や時間が増加する。次節以降で、要件および解決策を検討する。

2.3 要求事項

この節では、インターネット利用者を対象とした情報収集に必要な項目について検討する。本研究では、(1) 情報収集の迅速化、(2) 信頼性の高い回答データの取得と目標としている。そのために、以下のような要件を定義し、本研究の目的を検討する。

情報収集者の利便性

インターネットを利用することで、低コストで情報収集できるようになったが、個人で情報収集する場合には、CGI プログラムを書く、対象者に電子メールを送信するなど、まだ多くの手間と時間を要する。情報収集者は、質問票の作成や、質問票の配布、回答データの回収など、情報収集に要する作業をすべて自分で行なわなくてはならない。質問票に

[1] http://www.imi.ne.jp/imi/
[2] http://town.hi-ho.ne.jp/market/
[3] http://www.dik.co.jp/knots/
[4] http://www.gigajapan.co.jp/japanview.htm

記載される質問は、回答選択肢の中から一つ選択する単一回答形式、回答選択肢のうちから複数選択する複数回答形式、回答選択肢に順位をつける順位回答形式が大部分を占める。これらの代表的な質問の作成支援ツールが必要である。

インターネット上では、さまざまな情報収集対象があり、それに応じて情報収集手段がある。しかし、インターネット利用者を対象とした場合は、電子メールや WWW などの既存の情報交換プロトコルを組み合わせているのみで、情報収集の仕方もさまざまである。ゆえに、インターネット利用者を対象とした情報収集は、その手段がインターネット上でまだ確立されていないと言える。質問票の配布および回答データの回収のためのプロトコル、それを実現する情報交換システムが必要である。

回答者の利便性

本研究における情報収集は、回答者から回答データを得ることが目的であり、回答者の個人情報は必要としないので、必要以上の情報を情報収集者が取得できないようにする必要がある。回答者のプライバシが保護されなければ回答者数は減少する [3]。インターネット上で商品を購入しない一番の理由として、回答者のプライバシが守られているか不安を感じることがあげられている [1]。

しかしそのために回答者の利便性を損なっては、回答者は回答しない。回答者にランダムな文字列のパスワードを強要するなど、質問票の質問に回答する以外の作業を強要することは避けるべきである。電子メールを利用してアンケート調査に回答する場合には、回答者に回答データのフォーマットを崩さないよう強要する場合が多い。これも回答者の負担となる。また、回答しやすいインタフェースを用意する必要がある。計算機を利用すればさまざまな質問が作成でき、画像の利用 [4] など、回答しやすい質問の作成も必要である。

情報収集プロトコル

情報収集者の利便性の向上には情報収集のためのプロトコルが必要であると述べた。情報収集プロトコルには、情報収集者、回答者の利便性を損なわず、信頼性の高いデータを取得できることが要求される。以下に、インターネット利用者を対象とした情報収集プロトコルに必要な項目を挙げる。

機密性 第三者に通信内容が洩れないことを保証する。

完全性 通信内容がかいざんされていないことを保証する。

認証 第三者になりすまして回答できないことを保証する。

多重回答の防止 回答者は一回のみ回答できる。

匿名性 回答者情報と回答データの対応が取れない。回答者情報とは、回答者であることを示す識別子である。例えば以下のような情報が挙げられる。

- 電子メールアドレス
- 回答者の利用する計算機の IP アドレス

完全性、認証、多重回答の防止は、信頼性の高いデータを得るために必要な項目である。回答者のプライバシを保護するために、機密性と匿名性が必要である。

2.4 まとめ

本研究の目的を整理する。本研究は、インターネットを利用して、人々を対象とした情報収集を迅速化しつつ、より信頼性の高い情報を得ることを目的とする。そして、さまざまな情報収集に対応し、スケーラビリティを持つ情報収集機構を構築することを目的とする。そのために、要件の解決策を検討し、どのような情報収集機構が必要かを検討していく。

これまでに、筆者は卒業論文および文献 [5][6] において、「信頼性の高い回答データの取得」を実現するモデルとして、情報収集者と回答者の間に 2 つの機関を用意し、電子メールによる情報収集過程での回答者の発信元アドレスを秘匿する方法を考案し、実装してきた。今回、本研究内容を発展させ、より迅速かつ信頼性の高い情報収集モデルを確立する。

最後に、本論文では以下の用語を定義する。

収集者 情報を収集する人。

対象者 情報収集の対象者。本論文では、もっとも広い対象者はインターネット利用者となる。

回答者 対象者のうち、質問票に記載された質問に回答する人。

質問票 回答者が質問に回答するためのインタフェース。

回答データ 回答者の質問に対する回答。

集計データ ある一回の情報収集における回答データの集合。

第3章　現状の分析と解決策

この章では、全章で述べた情報収集プロトコルに必要な項目が、どの程度実現されているかを分析する。そして、インターネットに情報収集プロトコルを確立することの必要性を述べる。

3.1　インターネットのセキュリティ技術

この節では、現状のインターネットにおいて、機密性、完全性、認証、匿名性、多重回答の防止がどの程度実現されているかを分析する。

3.1.1　機密性・完全性・認証

機密性、完全性、認証は現在のインターネットで既に実現できる。機密性は通信内容の暗号化により実現できる。共有鍵暗号として DES、IDEA などが用いられ、公開鍵暗号として Diffie-Hellman、ElGamal、RSA などが用いられている。完全性は MD5、SHA-1 などの一方向ハッシュ関数を使用して、通信内容のハッシュ値を送信元と送信先で比較することにより実現できる。認証の方法はいくつかあるが、電子署名により実現できる。

これらを実現するアプリケーションには、SSL[7]、ssh[8]、pgp[9]、PEM[10][11][12][13]、S/MIME [14][15] などがある。どのアプリケーションも 2 者間の通信で利用される。通信は送信元の秘密鍵で署名され、送信先の公開鍵で暗号化される。これにより、機密性、完全性、相手認証を実現している。SSL は Netscape 社により規定が定められ、トランスポート層でこれらを実現している。ssh はヘルシンキ大学で開発された Shell でほとんどの UNIX システムに実装されている。UNIX でセキュリティ上問題になっていた r コマンド (rlogin, rsh, rcp など) に、認証と暗号化による安全な通信を提供する。pgp、PEM、S/MIME は電子メールのメッセージに機密性、完全性、相手認証を実現した。S/MIME はテキストだけでなく、MIME バイナリ文書にも適用できる。

しかし、これらの実装は匿名性を保持できない。これらの実装は 2 者間の通信であり、送信元の情報、受信したメッセージの内容ともお互いに知ることができる。インターネットでは、送信元の IP アドレスを隠蔽できない。ゆえに、情報収集者と回答者が直接通信しあう場合は、情報収集者に対する回答者の匿名性を保持できない。そのため匿名性を実現するには、何らかの中継者が必要となる。

3.1.2 匿名性

現在のインターネットで匿名性を実現する実装には、Proxy サーバ、Anonymizer[16]、Onion routing[17][18][19]、Crowds[20] などが挙げられる。以下でこれらの実装について述べる。

WWW クライアントが Proxy サーバを経由して、WWW サーバにアクセスすると、WWW サーバからは Proxy サーバからアクセスしてきたように見える。ゆえに、WWW サーバ管理者は WWW クライアントの情報を知ることができず、WWW サーバ管理者に対する WWW クライアントの匿名性が保持されていると言える。Anonymizer も Proxy サーバとほとんど同じ動作をする。Anonymizer のホームページに行き、参照したい WWW ページの URL を入力すればよい (図 3.1)。Anonymizer の方が Proxy サーバよりも WWW サーバ管理者に対して WWW クライアント利用者の個人情報を隠蔽できる。また、Proxy サーバの利用には、Proxy サーバと WWW クライアントが近い位置にある場合が多く、その場合は WWW クライアントを隠蔽できても WWW クライアントが存在する組織までは隠蔽できない。

しかし、どちらの実装も中継ホストの管理者である、Proxy サーバ管理者および Anonymizer 管理者に対しては WWW クライアントの匿名性を実現できない。また、どちらの実装も第三者の盗聴に弱く、第三者に対する匿名性も実現できない。情報収集者と回答者の間に情報収集代行者が介在するモデルもこれらの実装と同じであり、情報収集代行者に対して回答者の匿名性を保持できない。

Onion routing は MIX ネット [21] の理論を実現させたものである。Onion とは、各ルータの鍵で多重に暗号化されたデータが、ルータを経由するたびに復号化されること、すなわち暗号化の皮がむけていくことを意味する。Onion routing は、まず経路を確立するフェーズがあり、それから実際のデータを転送するフェーズがある。図 3.2 に動作例を示す。送信元ホスト A は ルータ X、Y をランダムに選び、送信先ホスト Z までの経路を確立する。A はメッセージ M を $M' = E_X(E_Y(E_Z(M)))$ と暗号化して X に送信する。E_Xは X との通信に用いる暗号化関数である。X は M'を復号化し、Y へ転送する。以下、Z までこの動作が繰り返される。この仕組みにより、Onion routing では中継ルータにもメッセージ M を秘密にできる。また送信先ホスト Z には送信元ホスト A を隠蔽できる。

Crowdsは、複数のホストから構成される。パケットはある確率で、目的のホストへ送信されるかまたはランダムに選んだ他のホストへ送信される (図 3.3)。メッセージは共有鍵暗号化方式で暗号化される。この仕組みにより、送信先のホストに送信元のホストを隠蔽している。

Onion routing、crowds は中継ホストに対しても送信元ホストの匿名性を保持できる。しかし、どちらのシステムも複数の計算機を用意しなければならず、実現が困難である。また、送信元ホストは送信先ホストまでの経路を通過するルータと通信するための鍵を保持している必要があり、どちらも閉じた空間のみにしか適用できない。匿名性の強度と情報収集者の利便性にはトレードオフの関係がある。また、これらの実装は匿名性を実現した結果、回答者の認証と多重回答の防止を実現できない。

図 3.1: anonymizer ホームページ

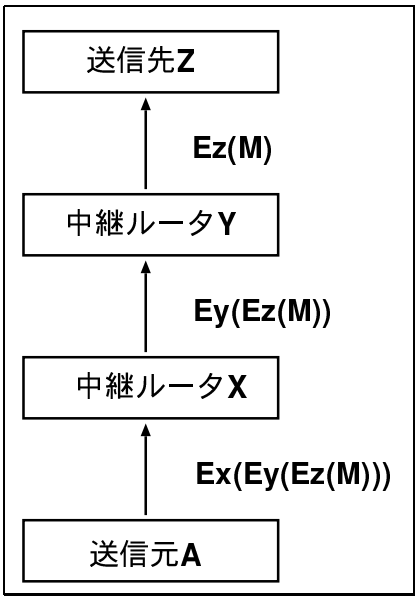

図 3.2: Onion routing

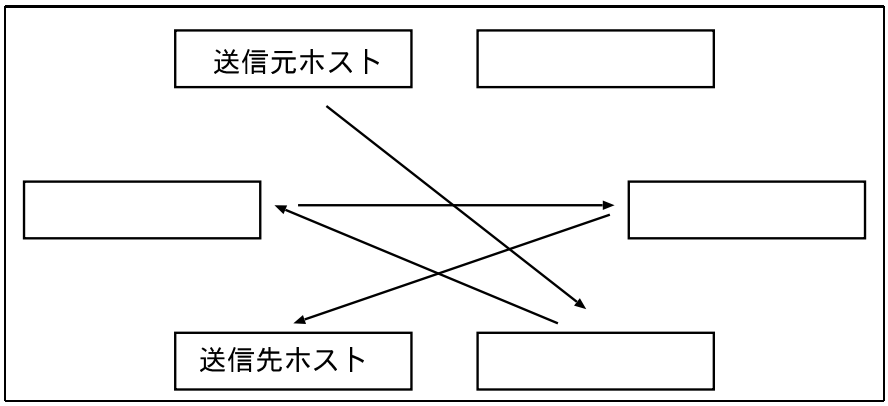

図 3.3: crowds

3.1.3　多重回答の防止

多重回答は送信元を認証することにより防止できる。しかし、現在のインターネットで、この機能が装備されているプロトコルは見当たらない。

3.2　関連研究

この節では、特に匿名性が重視されている電子選挙、電子決済の研究について述べる。

3.2.1　電子選挙に関する研究

現在、提案されている方式は以下のように分類できる [22][23]。

- MIX ネットを利用する方式
- マルチパーティープロトコルを利用する方式
- ブラインド署名を利用する方式

MIX ネット [21] は Chaum により提案され、投票者のプライバシを保護しつつ電子選挙を実現するための最初の解決策である。これをさらに発展させた方式も多く提案されている [24][25]。

以下では、MIX ネットを利用した電子選挙の概要を述べる。MIX ネットでは、複数のセンター $C_1, C_2, \cdots, C_n$ と掲示版 A を仮定する。掲示版 A にはセンターの公開鍵 $e_i (i = 1, \cdots, n)$ が公開されている。E_i を公開鍵 e_iでの暗号化関数とする。また e_iに対応する秘密鍵を d_iとし、復号関数を D_iとする。

1. 投票者 $V_j (j = 1, \cdots, k)$ は投票内容 v_jから

$$m_1 = E_1(E_2(\cdots(E_n(v_j))\cdots))$$

   を求める。そして m_1 に投票者 V_jの署名 $S_{V_j}(m_1)$ を付けてセンター C_1に送信する。
2. センター C_1は有権者名簿を持ち、$S_{V_j}(m_1)$ の正当性を検証し、正しい投票者であるか、未投票であるかを確認する。いずれも満たしている場合は、秘密鍵 d_i で復号化し、

$$m_2 = D_1(m_1) = E_2(\cdots(E_n(v_j))\cdots)$$

   をセンター C_2に送信する。
3. センター $C_i (i = 2, \cdots, n-1)$ について以下の操作を繰り返す。

$$m_{i+1} = D_i(m_i) = E_{i+1}(\cdots(E_n(v_j))\cdots)$$

   を計算し、 m_{i+1}をセンター C_{i+1}に送信する。

4. センター C_nは

$$D_n(m_n) = D_n(E_n(v_j)) = v_j$$

を得る。

投票内容はすべてのセンターを通過し、すべてのセンターが不正をしない限り、投票者の匿名性が保持される (図 3.4)。

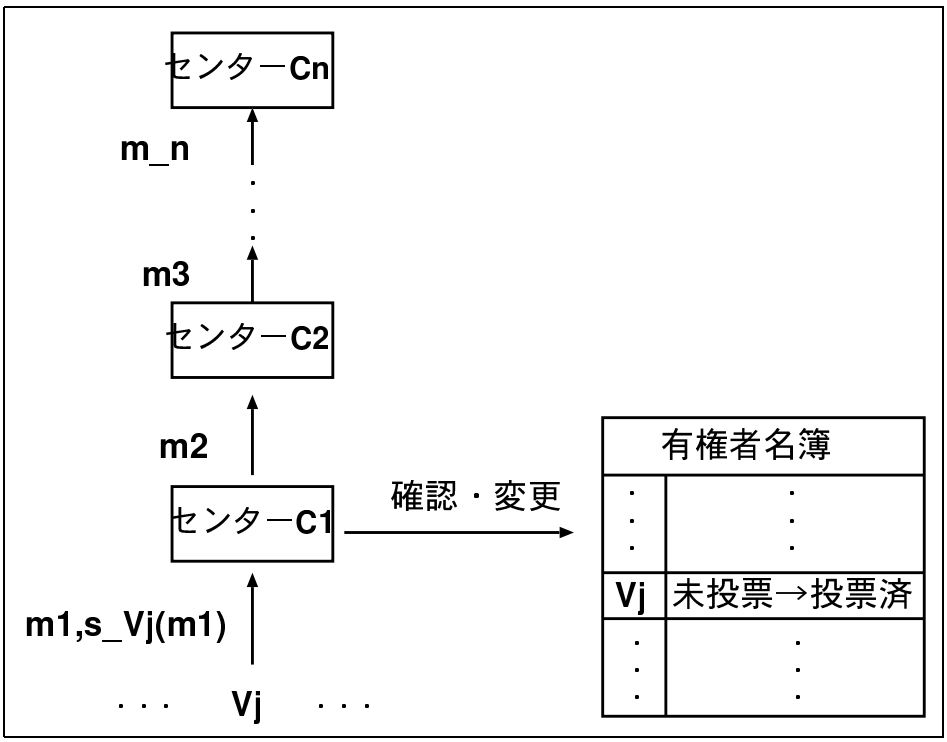

図 3.4: MIX ネット

マルチパーティープロトコルを利用した方式には文献 [26][27] などがある。以下ではマルチパーティープロトコルを利用した方式の概要を述べる (図 3.5)。この方式は高次剰余暗号を用いる [28]。この方式も複数のセンター $C_1, C_2, \cdots, C_n$ と掲示版 A を仮定する。掲示版 A にはセンターの公開鍵と $r(> k =$ 投票者数$)$ が公開されている。E_i をセンター C_iの公開鍵での暗号化関数とする。

1. 投票者 $V_j(j = 1, \cdots, k)$ は、投票内容 $v_j \in \{0, 1\}$ を

$$v_j = v_{j1} + v_{j2} + \cdots + v_{jn} \bmod r$$

と n 個に分割する。v_{ji} をセンター C_iの公開鍵で暗号化し、$E(v_{ji})$ をセンター C_iに送信する。
2. 各センター C_iは送信されてきた $E(v_{1i}), E(v_{2i}), \cdots, E(v_{ki})$ を復号化し、

$$M_i = v_{1i} + v_{2i} + \cdots + v_{ki} \bmod r$$

を得る。

3. 投票結果は

$$M_1 + M_2 + \cdots M_n \bmod r$$

で与えられる。

この方式も各センターに対して投票内容を秘密にできる。また投票結果からは投票者と投票内容を対応づけられない。ゆえに投票者の匿名性を保持できる。しかし、この方式は回答選択肢の数が 2 つである質問しかできない。質問票には、さまざまな形式の質問が記載されるので、この方式はインターネット利用者を対象とした情報収集には適用できない。

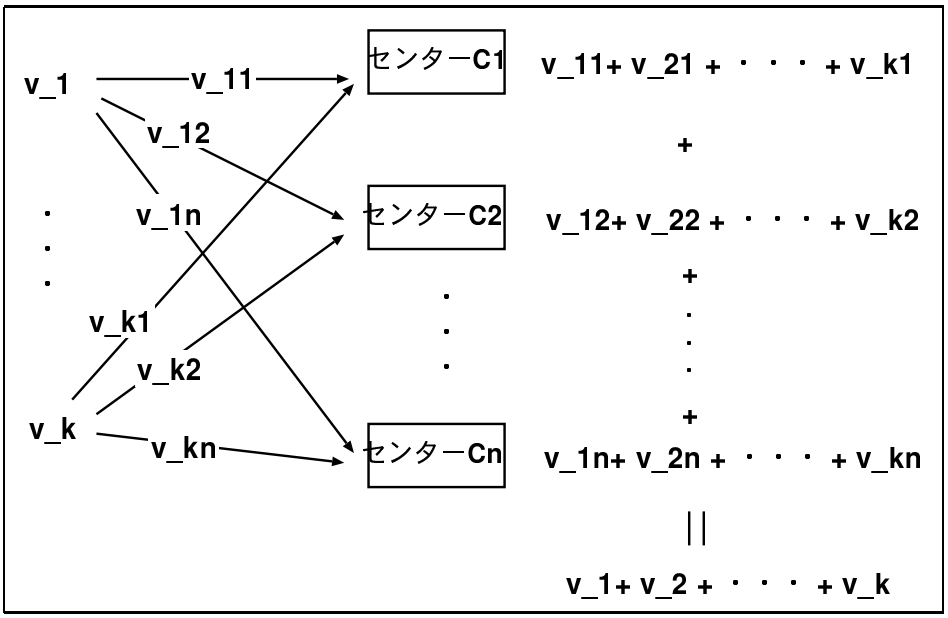

図 3.5: マルチパーティプロトコルを利用した方式

ブラインド署名 [29] は Chaum により提案され、文書の内容を秘密にして電子署名をする技術である。この技術を利用した電子選挙方法も提案されているる [30][31]。この方式では、認証者と集計者と投票者がいる。投票者は認証者にブラインド署名をしてもらう。そして、投票者は集計者に対し投票内容とブラインド署名を送信する。ブラインド署名からは認証者の情報はわかるが投票者の情報はわからない。しかし、この方法は MIX ネットや投票所 [23] などの匿名通信路を仮定している。

3.2.2 電子決済に関する研究

電子決済の方法は以下のように分類できる [32]。

- 小切手型
- クレジットカード型

- ジャストペイ型
- 電子マネー型

この節では、特にクレジットカード型および電子マネー型の匿名性の実現方法について述べる。

クレジットカード型はネットワーク上で商品を購入し、クレジットカードで決済する方式である。代表的な方式に、VISA[1] 社と MASTERCARD[2] 社が定めた SET[33] がある。

SET では、顧客、店舗、認証機関、クレジットカード会社がインターネットを介して接続されている。SET では、以下のような匿名性を実現する。

- クレジットカード番号はクレジットカード会社のみ知ることができる
- 購入した商品は店舗のみ知ることができる

以下でその実現方法について述べる[3] (図 3.6)。

1. 顧客は店舗に購入依頼データを送信する。
2. 店舗は顧客に、自分の公開鍵 A とクレジットカード会社の公開鍵 B を送信する。
3. 顧客は発注書データ H を店舗の公開鍵 A で暗号化し、クレジットカード番号 C をクレジットカード会社の公開鍵 B で暗号化して、$E_A(H)$ と $E_B(C)$ を店舗に送信する。ただし E_A, E_Bは公開鍵 A,B での暗号化関数とする。
4. 店舗はクレジットカード会社に $E_B(C)$ を送信し、振込を依頼する。
5. クレジットカード会社は店舗に振込結果を送信する。

このように、店舗が購入した商品情報を保持し、クレジットカード会社がクレジットカード番号を保持することで、購入した商品情報とクレジットカード番号を保持する機関を分離した。この仕組みにより匿名性を実現している。さらに通信内容はすべて暗号化されているので、第三者にこれらの情報が洩れることはない。

次に、電子マネー型について述べる。電子マネー型はさらにカード型とネットワーク型に分類できる。ここでは、ネットワーク型である、Digicash 社[4] の E-cash を取りあげる。

E-cash では、利用者、銀行、店舗間の通信により、電子決済がなされる。E-cash では銀行印が押された電子マネーを使用できる。電子マネーには固有の番号 (札番号) がつけられている。E-cash における匿名性は、札番号から利用者がわからなようにすることを指す。

この実現には、ブラインド署名 [29] が利用されている。利用者は札番号を決め、銀行に札番号を秘密にして、銀行印を押してもらう。これにより、銀行は札番号から利用者が誰かわからない。そして利用者は銀行印が押された電子マネーを使用して買物をする。

[1] http://www.visa.com
[2] http://www.mastercard.com
[3] ここでは匿名性に関する処理のみについて述べているため、実際にはもう少し複雑な処理がなされる
[4] http://www.digicash.com

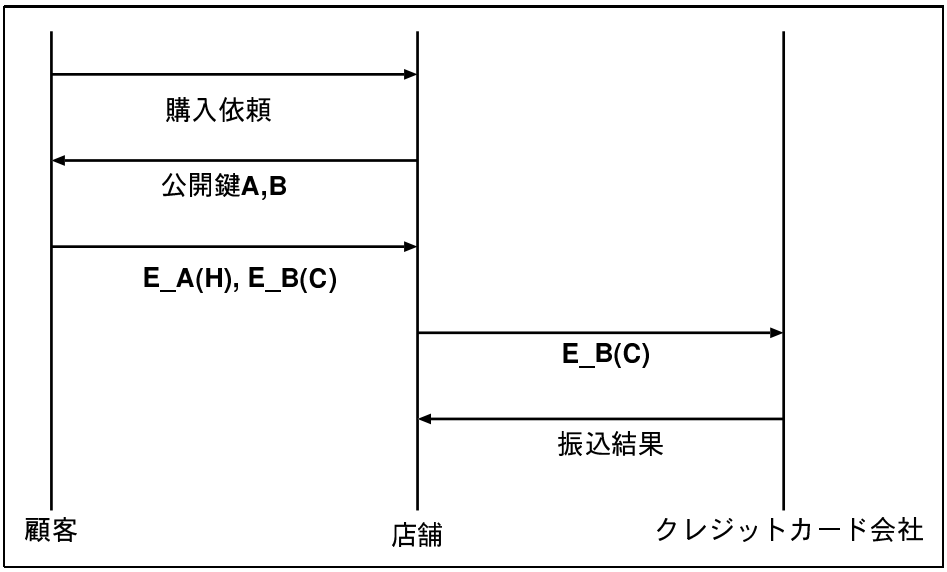

図 3.6: SET でのデータフロー

3.3 本研究のアプローチ

現在での情報収集の典型的な例は、電子メールで質問票または質問票の URL を配布し、電子メールまたは WWW 上の質問に回答してもらう方法である。これらの情報収集は、2.3 節で挙げた要件を満たしていない。

現在のインターネットでは、完全性、機密性、認証は、既に実現するための技術が確立されている。そして完全性、機密性、認証を実現するプロトコルも多く実装されている。匿名性に関しては、電子選挙に関する研究において、実現方法が議論されているが、まだインターネット上に実装されるまでには至っていない。多重回答の防止は認証の技術を用いれば容易に実現できる。

以上のことから、情報収集のためのプロトコルに必要な条件を実現する個々の技術については、多く議論され、インターネット上に実装されつつある。しかし、現在のインターネットでは、要件を全て満足するプロトコルはまだ見当たらない。

そこで本研究では、まず人々を対象とした情報収集のためのプロトコルをインターネット上に確立する。このプロトコルにより、信頼性の高いデータを取得する。そして、そのプロトコル上に情報収集者および回答者のユーザインターフェイスを構築し、簡単かつ迅速な情報収集を目指す。

第4章　rics の設計と実装

この章ではインターネット利用者を対象とした情報収集機構 rics の設計と実装について述べる。そして、信頼性の高いデータを得るための仕組み、情報収集者および回答者の利便性を向上するための仕組みを明らかにする。

4.1　rics の機能

rics はインターネット利用者を対象として情報収集するシステムである。rics は情報収集者、回答者の利便性を重視し、簡単かつ迅速に情報収集できる機構を提供する。質問票の作成、質問票の配布、回答データの回収など、情報収集に要する大部分の作業を自動化している。情報収集者は質問内容を決め、rics に情報収集を依頼するのみでよい。

rics は 3 つの階層を持つ。TCP/IP 層の上位層に rics プロトコル層があり、その上位層に rics データフローモデル層、rics アプリケーション層がある (図 4.1)。それぞれの階層ごとに役割が規定されている。rics プロトコルが通信の機密性、完全性を実現し、rics データフローモデルが回答者の匿名性、認証、多重回答の防止を実現する。rics アプリケーションは、情報収集者と回答者へのユーザインターフェイスを提供する。

図 4.1: rics の階層構造

4.2 rics の動作原理

rics では、情報収集者と回答者の間に集計サーバと情報収集サーバが介在する (図 4.2)。インターネット上に集計サーバと情報収集サーバが配置される。集計サーバと情報収集サーバは異なる管理者により管理される。

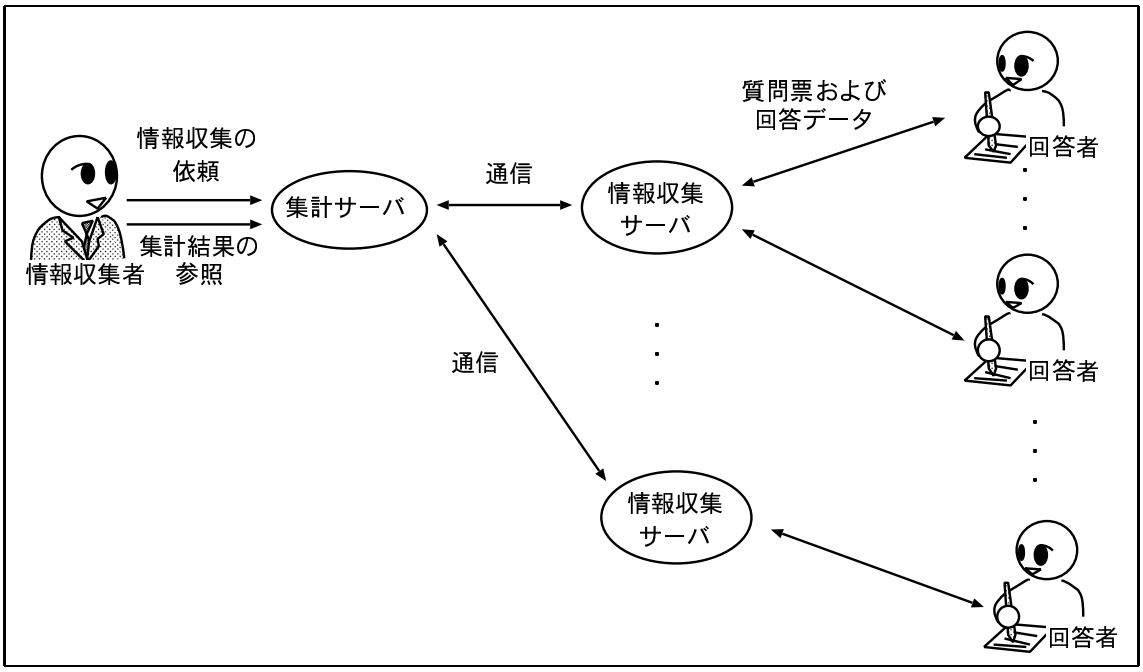

図 4.2: rics の計算機構成

集計サーバは情報収集者の役割を果たす。集計サーバは情報収集者の作業量軽減のために用意されたサーバである。通常、集計サーバは情報収集者の利用する計算機で起動される。集計サーバは情報収集者から情報収集の依頼を受け、情報収集サーバに対し情報収集を依頼する。そして、情報収集サーバから回答データを回収する。簡単な集計をし、集計結果を提示する。

情報収集サーバは、対象者に対し質問票を配布し、回答者からの回答データを回収する。そして、回答データを集計サーバに送信する。情報収集サーバは各組織ごとに運用される。

4.3 rics プロトコル

rics プロトコルは、rics アプリケーション間の通信プロトコルである。rics プロトコルによる通信は、最初に接続元と接続先の公開鍵を交換する。その後の通信はすべて送信元の秘密鍵で署名され、送信先の公開鍵で暗号化される。これにより通信の機密性と完全性が実現される。rics プロトコルは目的に応じて rics-send1 プロトコル、rics-send2 プロトコル、rics-collect1プロトコル、rics-collect2 プロトコルがある (図 4.3)。

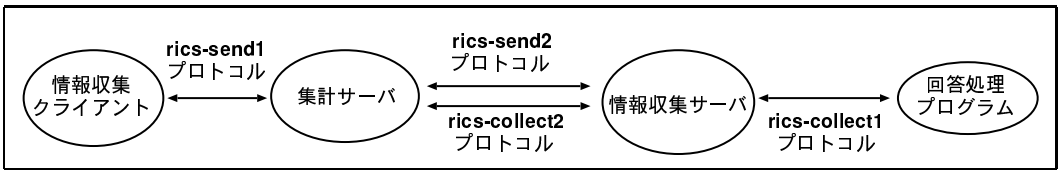

図 4.3: rics プロトコルの分類

4.4 rics データフローモデル

rics データフローモデルは質問票と回答データの流れを定義している。この層では匿名性、認証、多重回答の防止を実現する。以下に、質問票と回答データの流れを示す (図 4.4)。集計サーバの公開鍵を k、質問票を X、回答データを Y とする。

1. 情報収集者は X を集計サーバに送信する。
2. 集計サーバは X と k を情報収集サーバに送信する。
3. 情報収集サーバは X と k を回答者に送信する。
4. 回答者は Y を作成し、同時に Y を k で暗号化する。暗号化関数を E_k とし、暗号化されたデータを $E_k(Y)$ とする。
5. 回答者は $E_k(Y)$ を情報収集サーバに送信する。
6. 情報収集サーバは $E_k(Y)$ を集計サーバに送信する。
7. 集計サーバは $E_k(Y)$ を集計サーバの秘密鍵で復号化して Y を得る。
8. 集計サーバは情報収集者に Y を送信する。

rics データフローモデルにより、情報収集サーバは回答データ Y を知り得ない。また、情報収集サーバは回答者情報を集計サーバに送信しないので、集計サーバは回答者情報を知り得ない。この仕組みにより回答者の匿名性を保持する。表 4.1 に集計サーバ、情報収集サーバ、回答者が知り得る情報をまとめる。〇が情報を知り得ることを指し、×が情報を知り得ないことを指す。

	回答者情報	回答データ
集計サーバ	×	〇
情報収集サーバ	〇	×
回答者	〇	〇

表 4.1: 匿名性

rics データフローモデルは MIX ネット方式のモデルに近い。MIX ネット方式と異なる点は、データの流れが双方向であることである。rics データフローモデルでは、質問票の配布を定義した。これにより、集計サーバの公開鍵を回答者に送信できた。MIX ネット方式では、データの流れは回答者から情報収集者への片方向であった。そのため、中継セン

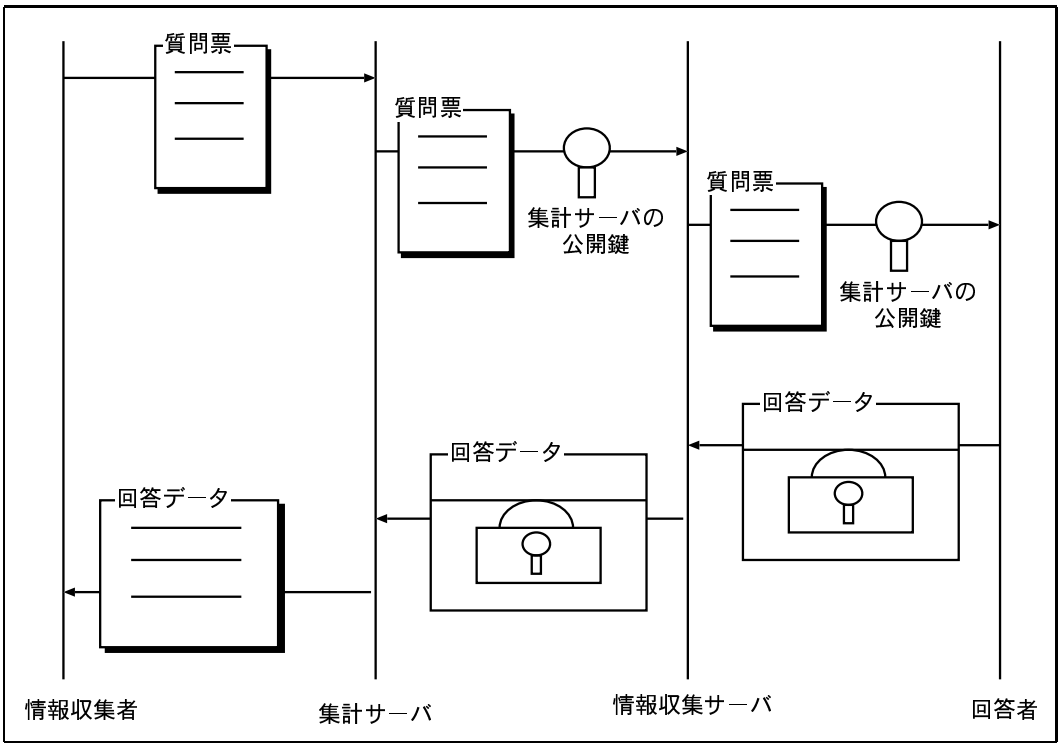

図 4.4: データの流れ

ターの公開鍵を公開しておくための掲示版が必要となった。rics データフローモデルでは集計サーバの公開鍵を質問票と一緒に回答者に送信することにより、用意する機関を必要最低限に押え、実現可能性を向上させた。

回答者の認証は、情報収集サーバが回答データを受理するときに行う。同時に、2 度以上回答できないようにする。

4.5 rics アプリケーション

この節では、rics アプリケーションの動作について述べる。rics アプリケーションは情報収集者および回答者へのユーザインターフェイスを提供する。そして、rics アプリケーション間の通信により、情報収集がなされる。

4.5.1 ユーザインターフェイス

rics アプリケーションには以下の 4 つのアプリケーションがある。

- 情報収集クライアント
- 集計サーバ

- 情報収集サーバ
- 回答処理プログラム

情報収集クライアントは、情報収集者が集計サーバと通信するためのインターフェイスとなる。情報収集者は情報収集クライアントを利用して、集計サーバに対し情報収集を依頼する。

回答処理プログラムは、回答者が情報収集サーバと通信するためのインターフェイスである。回答者は回答処理プログラムを利用して、情報収集サーバに回答データを送信する。

rics を利用した情報収集では、質問票を回答者に提示し、集計結果を情報収集者に提示する。この 2 つのデータはユーザ[1] の環境に適した形で提供される。現在では、質問票は WWW ブラウザ (図 4.9)、IBM ChipCard VW200 (図 4.18) から参照できる。集計結果は WWW ブラウザから参照できる (図 4.8)。

4.5.2 データの表現形式

質問票および集計結果のデータ形式はユーザの環境に依存する。情報収集者が環境依存のファイルを作成するのは、その環境ごとのデータ形式を覚えなければならず、情報収集者の負担となる。rics では、ユーザの環境に依存しない質問票定義コードと集計データを定義している。それぞれのデータの関係は図 4.5 で表される。

質問票定義コードは情報収集者により作成される。質問票定義コードも情報収集クライアントを利用して容易に記述できるようになっている。情報収集サーバが質問票定義コードから質問票を作成する。回答者が質問票に記載された質問に回答すると、回答処理プログラムにより回答データが作成される。集計サーバが集計データと質問票定義コードから集計結果を作成する。

質問票定義コードおよび集計データは XML[34] で記述され、その文法は DTD として定義される。XML は、大野研究室[2] [3] におけるシステム管理情報の標準形式として採用され、現在その標準形式を扱うためのシステムの開発が進んでいる。rics 内部文書の DTD は上記のシステムが扱う標準形式のサブセットとして定義されており、容易にこのシステムと連携できると考えられる。例えば、さまざまなビューアで集計結果を参照でき、さまざまな方法で集計データを解析できる。

質問票定義コード 質問票定義コードの DTD は `questionnaire.dtd` というファイル名で保持される (図 4.6)。質問票定義コードには質問票を作成するために必要な項目が記述される。`item` タグで一つの質問を記述する。現在では単一回答形式、複数回答形式、自由回答形式、アナログ回答形式が提供されている。`text` タグで質問票に中に表示する文章を記述する。質問票定義コードの詳細についてはついては付録 A で述べる。

[1] 本論文では回答者と情報収集者を合わせてユーザと呼ぶ
[2] 東京工業大学 情報理工 学研究科
[3] http://www.ohnolab.org

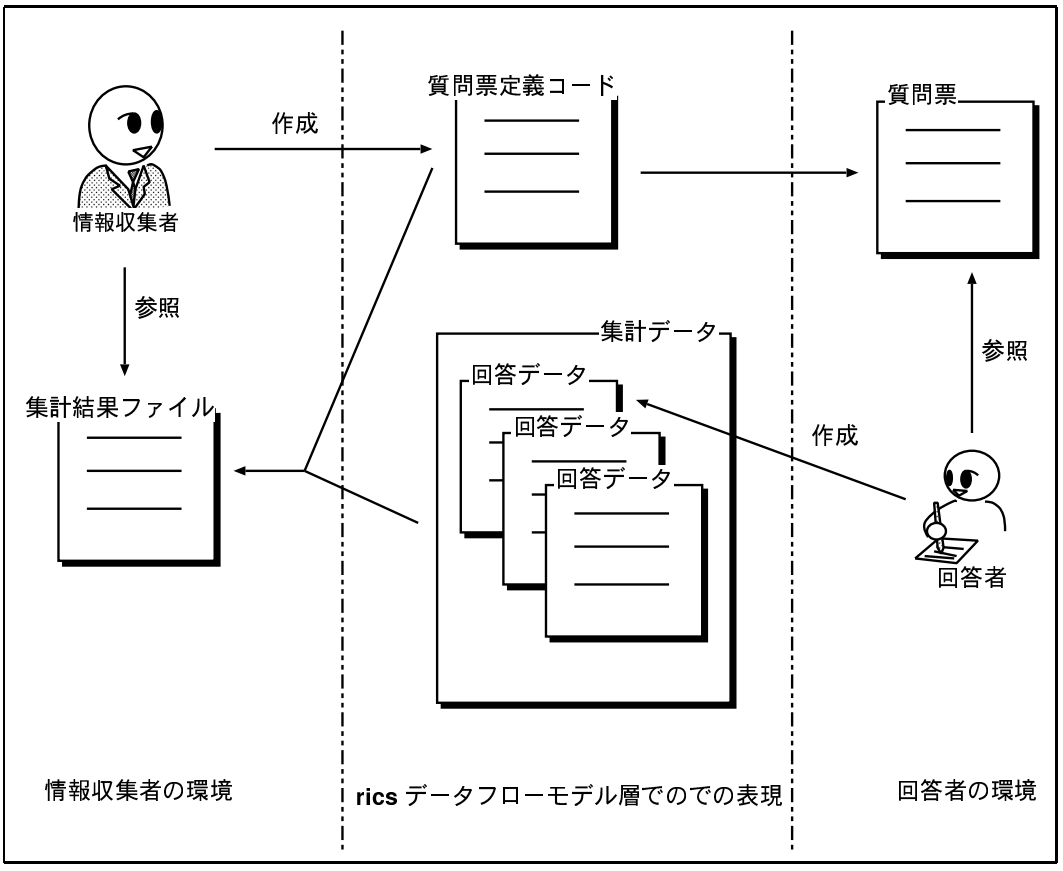
質問票定義コード
作成
質問票
情報収集者
参照
集計データ
回答データ
回答データ
回答データ
集計結果ファイル
参照
作成
回答者
情報収集者の環境
rics データフローモデル層でのでの表現
回答者の環境

図 4.5: データの関係

```
<!ELEMENT questionnaire (title, (item|text)*, exec?)>

<!ELEMENT title (#PCDATA)>
<!ELEMENT item (simple_answer|multiple_answer|free_answer|analog_answer)>
<!ATTLIST item id CDATA #REQUIRED>
<!ELEMENT text (#PCDATA)>
<!ELEMENT exec (#PCDATA)>

<!ELEMENT simple_answer (question, choice*)>
<!ELEMENT multiple_answer (question, choice*)>
<!ELEMENT free_answer (question, blank)>
<!ELEMENT analog_answer (question, left, right, length)>

<!ELEMENT question (#PCDATA)>
<!ELEMENT choice (#PCDATA)>
<!ELEMENT blank EMPTY>
<!ATTLIST blank width CDATA #REQUIRED
               height CDATA #REQUIRED>
<!ELEMENT left (#PCDATA)>
<!ELEMENT right (#PCDATA)>
<!ELEMENT length (#PCDATA)>
```

図 4.6: 質問票定義コード DTD

集計データ 集計データは回答データの集合である。集計データの DTD は`answers.dtd`というファイル名で保持される (図 4.7)。`answer` タグが回答データである。回答データには回答者の回答内容が含まれる。`item` タグで中に一つの質問に対する回答内容が記述される。

```
<!ELEMENT answers (answer*)>

<!ELEMENT answer (item*)>

<!ELEMENT item (simple_answer|multiple_anwer|free_answer|analog_answer)>
<!ATTLIST item id CDATA #REQUIRED>

<!ELEMENT simple_answer (selected)>
<!ELEMENT multiple_answer (selected*)>
<!ELEMENT free_answer (text)>
<!ELEMENT analog_answer (value)>

<!ELEMENT selected (#PCDATA)>
<!ELEMENT text (#PCDATA)>
<!ELEMENT value (#PCDATA)>
```

図 4.7: 集計データ DTD

4.6 実装

この節では、rics アプリケーションの実装について述べる。

4.6.1 集計サーバ

集計サーバは情報収集クライアントからの接続を受け付ける。集計サーバは、まず情報収集 ID を作成する。情報収集 ID は質問票定義コードおよび集計データを識別するためのものである。次に、rics-send2 プロトコルを用いて情報収集サーバに必要な項目を送信する。集計サーバには通信する情報収集サーバをあらかじめ登録しておく。複数の情報収集サーバを登録できる。最後に、集計結果の URL を情報収集クライアントに送信する。

また、集計サーバは情報収集サーバからの接続を受け付ける。rics-collect2 プロトコルにより回答データと情報収集 ID を受け取る。集計サーバは、指定された情報収集 ID の集計データに受け取った回答データを追加する。次に集計サーバは、XML パーサーを用いて質問票定義コードと集計データを解析し、簡単な集計をして、その結果を集計結果ファイルに保存する (図 4.8)。集計結果ファイルは集計データが更新されるたびに更新される。ゆえに、情報収集者はリアルタイムで集計結果がわかる。また、情報収集者自身も XML

で記述された集計データを得ることができ、簡単な集計だけでなく、さまざまな解析が可能となる。

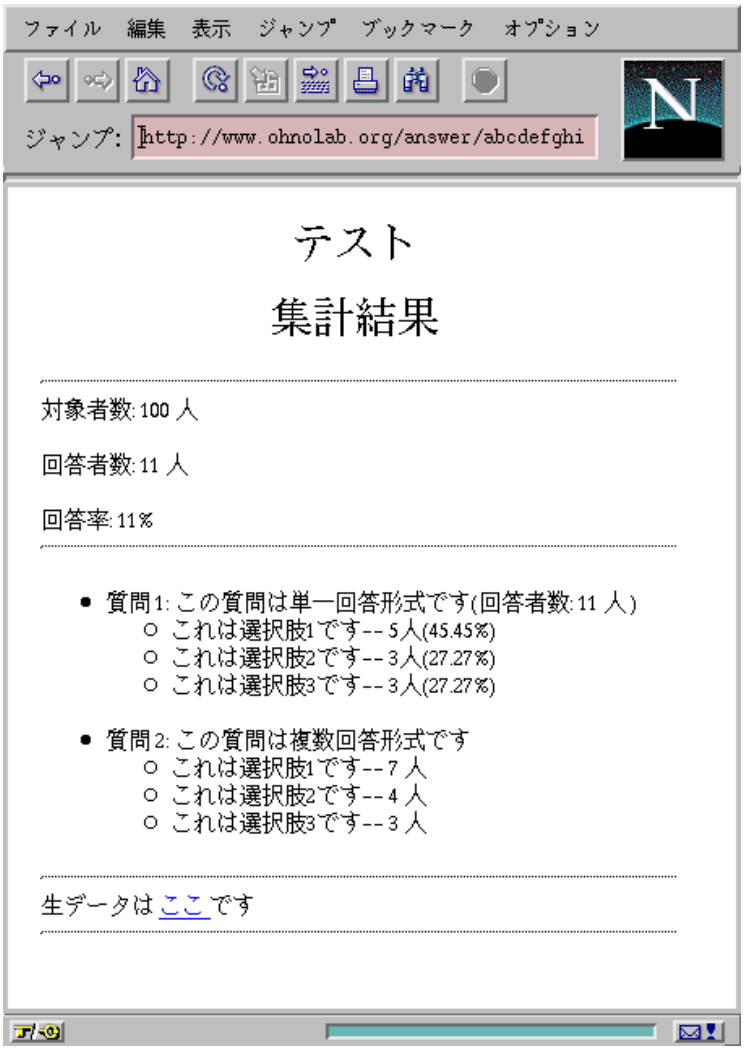

図 4.8: 集計結果ファイルの参照

4.6.2 情報収集サーバ

情報収集サーバは集計サーバからの接続を受け、対象者から情報収集する。信用できる集計サーバからの接続のみを許可するように設定できる。情報収集サーバはデータベースを持つ。データベースには、対象者の公開鍵など、対象者に質問票を配布するために必要な情報が記述されている。

情報収集サーバは XML パーサーを用いて質問票定義コードを解析し、質問票と回答処理プログラムを作成する。質問票および回答処理プログラムは複数の環境に対応できる。

現在は、WWW を利用した実装が実験および運用されているが、CITRUS[35] を利用した実装も進んでいる。CITRUS との連携については 4.7 節で述べる。

WWW を利用する場合には、質問票は HTML で記述される。対象者に対し、質問票の URL を対象者に通知する。回答者は情報収集サーバから通知された URL で質問票を参照する (図 4.9)。

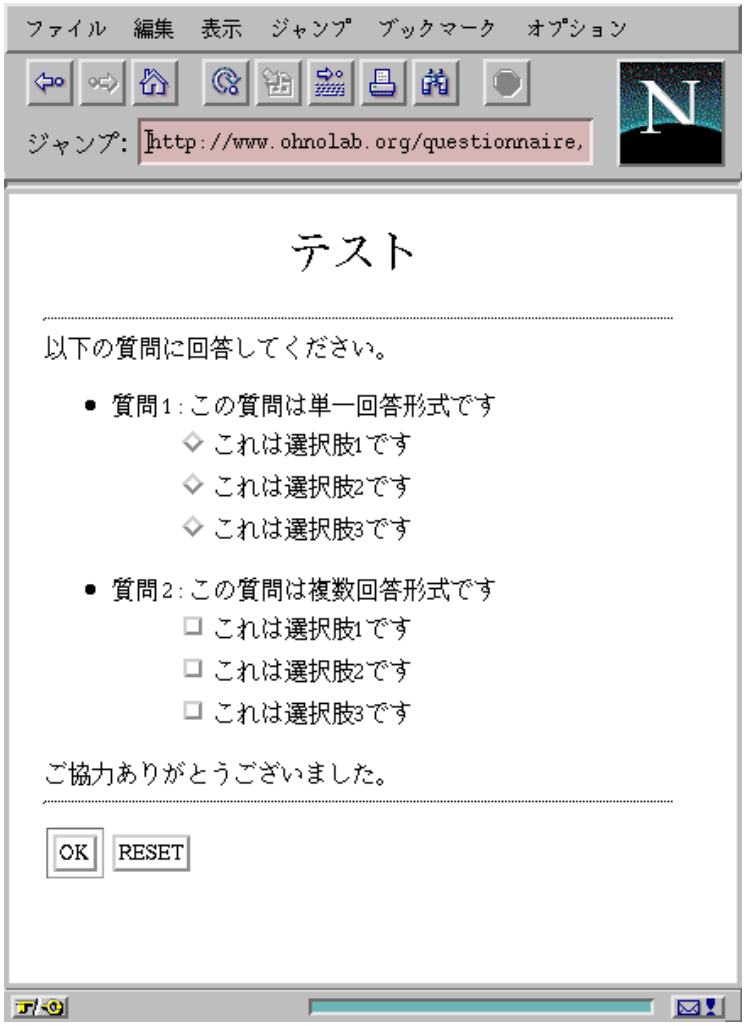

図 4.9: WWW ブラウザを使用した質問票のブラウジング

情報収集サーバは、質問票の URL を以下の 2 通りの方法で通知し、状況に応じて使い分ける。

- 電子メール
- 質問票 BOX

電子メールを利用する場合は、情報収集サーバのデータベースには対象者の電子メールアドレスが登録されている。情報収集サーバは、対象者に対し電子メールで質問票の URL

を送信する (図 4.10)。質問票の URL は対象者ごとに異なる。回答者はその URL を WWW ブラウザを使用して参照する (図 4.9)。

```
Jun-ichi SAKAI <sakai@ohnolab.org> さんから情報収集の依頼です。

以下の質問票の質問に回答して下さい。

タイトル: これはタイトルです
URL: http://www.ohnolab.org/questionnaire/abcdefghijklmnop.html

よろしくお願いします。
```

図 4.10: 回答者に送信される電子メール

質問票 BOX とは、WWW 上にその個人のみが参照できるページを指す (図 4.11)。この方法を利用する場合は、情報収集サーバのデータベースには対象者の質問票 BOX の URL が登録されている。回答者は自分の質問票 BOX を参照することにより、自分に届いた質問票を参照できる。質問票の参照は WWW ブラウザを使用する (図 4.9)。

以下では、回答者の認証について述べる。情報収集サーバは、質問票を対象者ごとに作成し、質問票の URL を対象者のみに通知する。電子メールを利用する場合には、質問票の URL を対象者の公開鍵で暗号化し、情報収集サーバの秘密鍵で署名して対象者に送信する。質問票 BOX を利用する場合には、あらかじめ安全な方法で質問票 BOX の URL を対象者に通知しておく。例えば、オフラインで通知するかまたは、質問票 BOX の URL を対象者の公開鍵で暗号化し、情報収集サーバの秘密鍵で署名して電子メールで送信する。

URL はランダムな 16 文字以上の文字列で構成されており、第三者が推測することは困難である。ゆえに、ある一つの URL を知り得るのは対象者本人のみである。この仕組みにより、情報収集サーバは質問票を参照できるのは対象者本人のみであると仮定して回答者を認証している。そして、ある URL からの回答を一回のみ受理している。

4.6.3 回答処理プログラム

回答処理プログラムは情報収集サーバにより作成される。その際、回答処理プログラムに集計サーバの公開鍵を保持させる。回答処理プログラムは JAVA アプレットコードである。すなわち、回答処理プログラムが質問票の実体であると言える。回答処理プログラムは、回答データを暗号化し、rics-collect2 プロトコルを用いて暗号化した回答データを情報収集サーバに送信する。

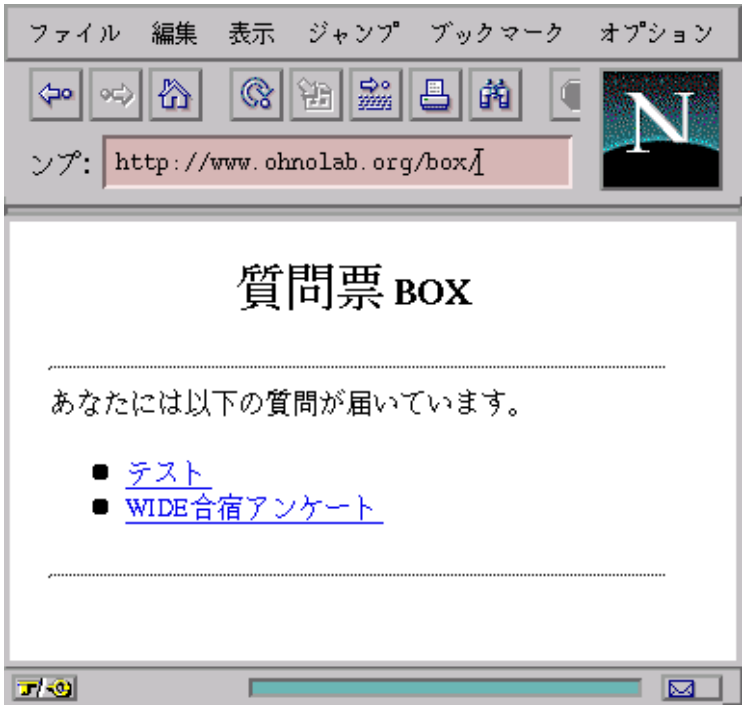

図 4.11: 質問票 BOX

4.6.4 情報収集クライアント

情報収集者は情報収集クライアントを利用して集計サーバに必要な項目を送信する。情報収集クライアントは rics-send1 プロトコルを用いて、集計サーバに対して情報収集に必要な項目を送信する。情報収集クライアントは情報収集者の環境に適したものが用意されるべきである。質問票定義コードを記述するためのエディタが用意されていれば、さらに情報収集者の負担が軽減する。

現在では、情報収集クライアントとして`ClientCom` クラス、メールクライアントが用意されている。

図 4.12 に`ClientCom` クラスの利用例を示す。実行後、集計結果の URL が表示される。そして、情報収集者はその URL を WWW ブラウザを利用して参照し、集計結果を得る (図 4.8)。

メールクライアントを利用する場合は、例えば図 4.13 の内容を電子メールで送信することにより、集計サーバと通信できる。またメールクライアントには質問票定義コードを C 言語風に記述できるものも用意されている (図 4.14)。メールクライアントは、情報収集者に対し電子メールで集計結果の URL を送信する (図 4.15)。

```
> cat data.txt
<?xml version="1.0" ?>
<!DOCTYPE questionnaire SYSTEM "questionnaire.dtd">

<questionnaire>

<title> テスト </title>

<text> 以下の質問に回答してください。 </text>

<item id="q1">
<simple_answer>
<question> この質問は単一回答形式です </question>
<choice> これは選択肢 1 です </choice>
<choice> これは選択肢 2 です </choice>
<choice> これは選択肢 3 です </choice>
</simple_answer>
</item>

<item id="q2">
<multiple_answer>
<question> この質問は複数回答形式です </question>
<choice> これは選択肢 1 です </choice>
<choice> これは選択肢 2 です </choice>
<choice> これは選択肢 3 です </choice>
</multiple_answer>
</item>

<text> ご協力ありがとうございました。 </text>

</questionnaire>
> java ClientCom -help
Usage: java ClientCom -d <questionnaire-define-code> -f <your name>
 -m <method> -h <shukei-server-hostname> [-g <group>] [-a | -q]
> java ClientCom -f sakai@ohnolab.org -m a -g ohnolab -d data.txt \
 -h localhost -a
http://www.ohnolab.org/rics/answer/abcdefghijklmnop.html
```

図 4.12: ClientCom クラスの利用

```
From: Jun-ichi SAKAI <sakai@ohnolab.org>
To: rics@ohnolab.org

METHOD: a
GROUP: ohnolab

<?xml version="1.0" ?>
<!DOCTYPE questionnaire SYSTEM "questionnaire.dtd">

<questionnaire>

<title> テスト </title>

<text> 以下の質問に回答してください。 </text>

<item id="q1">
<simple_answer>
<question> この質問は単一回答形式です </question>
<choice> これは選択肢 1 です </choice>
<choice> これは選択肢 2 です </choice>
<choice> これは選択肢 3 です </choice>
</simple_answer>
</item>

<item id="q2">
<multiple_answer>
<question> この質問は複数回答形式です </question>
<choice> これは選択肢 1 です </choice>
<choice> これは選択肢 2 です </choice>
<choice> これは選択肢 3 です </choice>
</multiple_answer>
</item>

<text> ご協力ありがとうございました。 </text>

</questionnaire>
```

図 4.13: メールクライアントの利用

```
From: Jun-ichi SAKAI <sakai@ohnolab.org>
To: rics@ohnolab.org

METHOD: a
GROUP: ohnolab

title = "テスト";

define simple q1 {
    question = "この質問は単一回答形式です";
    choice = "これは選択肢 1 です";
    choice = "これは選択肢 2 です";
    choice = "これは選択肢 3 です";
}

define multiple q2 {
    question = "この質問は複数回答形式です";
    choice = "これは選択肢 1 です";
    choice = "これは選択肢 2 です";
    choice = "これは選択肢 3 です";
}

main
{
    print("以下の質問に回答してください。");
    qprint(q1);
    qprint(q2);
    print("ご協力ありがとうございました。");
}
```

図 4.14: メールクライアントの利用 (C 言語風)

```
Subject: [Questionnaire] Your request was sent to rics.
From: rics-request@ohnolab.org
To: Jun-ichi SAKAI <sakai@ohnolab.org>

情報収集を開始しました。
集計結果は以下の URL を参照してください。

http://www.ohnolab.org/rics/answer/abcdefghijklmnop.html
```

図 4.15: メールクライアントからの応答

4.7 CITRUSとの連携

質問票の配布、回答データの回収に CITRUS を利用する実装も進んでいる (図 4.16)[36][37]。CITRUS は学内情報サービスを目的とした、公衆情報端末と超小型携帯端末を連携させた情報交換システムである。公衆情報端末として PICKLES[38][39] 端末、超小型携帯端末として ChipCardVW200 (以下 ChipCard と略す) を使用している。PICKLES 端末は大野研究室によって開発され、時間や場所の制約を受けずにインターネット上のサービスを利用できる環境の構築を目的としている。

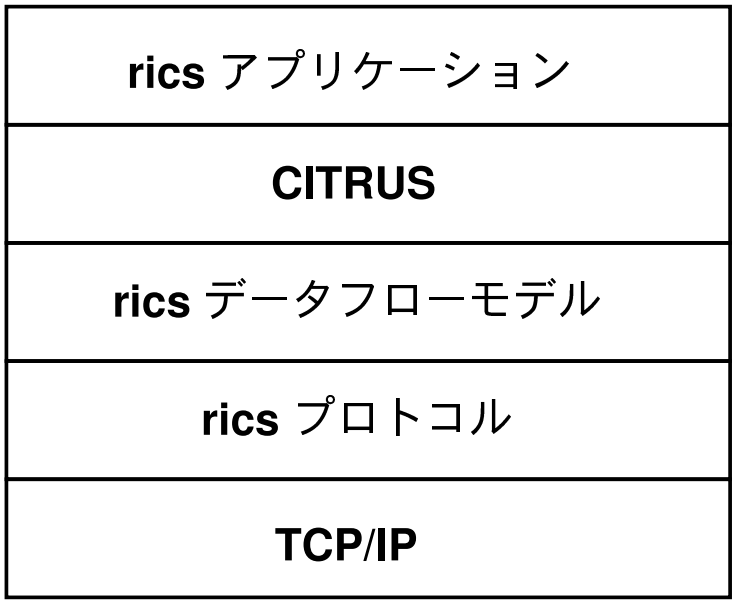

図 4.16: CITRUS の利用

CITRUS を利用したときの質問票の配布および回答データの回収の流れは以下のようになる (図 4.17)。

1. 情報収集サーバから PICKLES 端末へ質問票が送信される。
2. PICKLES 端末中の質問票を ChipCard に受信する。
3. 質問票を ChipCard に出力する (図 4.18)。
4. 回答を ChipCard に記録する。
5. ChipCard から PICKLES 端末へ回答データを送信する。
6. PICKLES 端末から情報収集サーバへ回答データを送信する。

現在、2,3,4,5 の部分が実装済で、1,6 の部分は実装中である。

回答者の認証は、ChipCard を使用できるのは本人のみであると仮定して認証している。この仮定は、文献 [37] での認証機構により実現されている。

以下で、この認証機構について述べる。本来の ChipCard 使用者があらかじめ定めておいた暗証番号のハッシュ値を ChipCard に記録しておく。ChipCard に質問票を表示する際

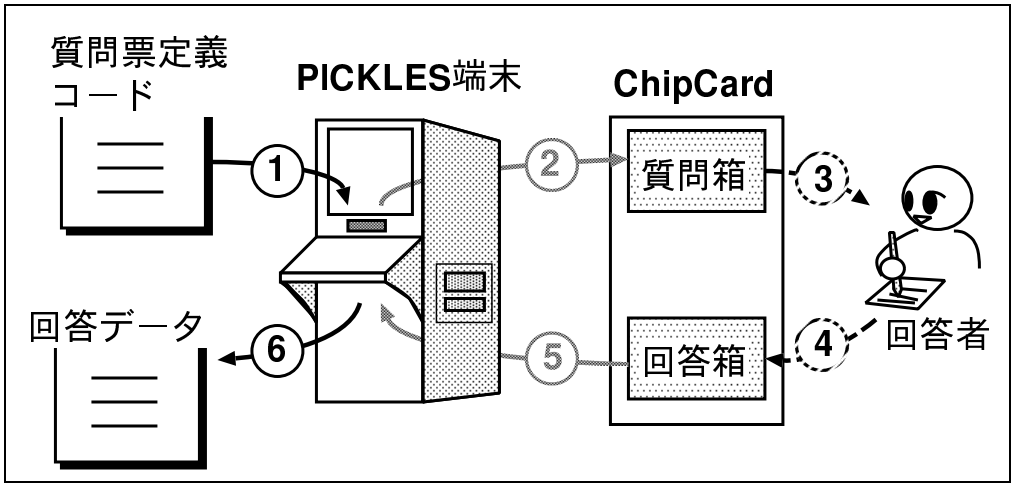

図 4.17: CITRUS を利用したときの流れ

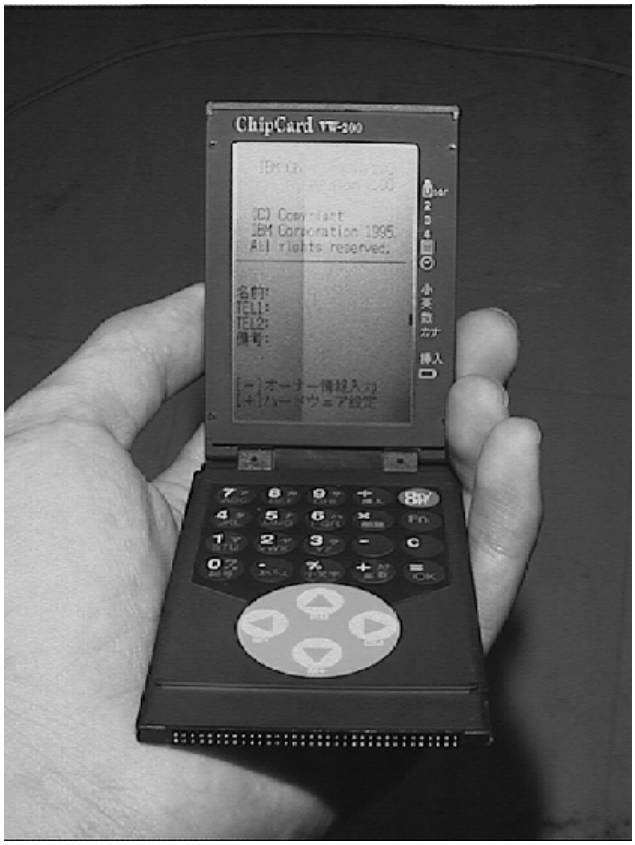

図 4.18: ChipCard による質問票のブラウジング

には、まず暗証番号を入力する。ChipCard は入力された暗証番号のハッシュ値を計算し、その値とあらかじめ内部メモリに記録されていた値を比較する。この 2 つの値が一致すれば質問票が表示され、質問に回答できる。

第5章　rics の安全性に関する評価と考察

この章では、まずインターネットにおける信頼モデルと公開鍵保証モデルを定義する。そして、情報収集形態を分類し、情報収集形態ごとの安全性について述べる。

5.1　信頼モデルと公開鍵保証モデル

rics を利用して情報収集するには、まず、情報収集者、対象者、集計サーバ、情報収集サーバが適切な信頼関係を持つ必要がある。この信頼関係には何通りもの信頼関係が考えられる。例えば、図 5.1 のような信頼関係がある。この章では、ある一つの信頼関係を、信頼モデルと定義する。

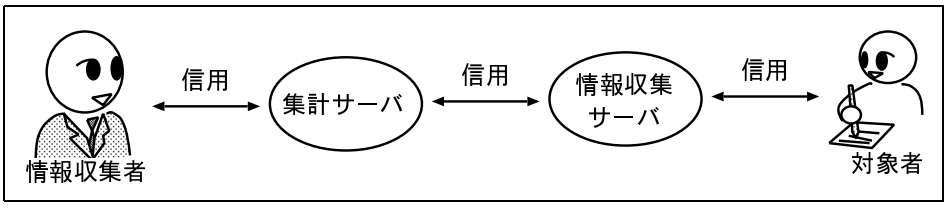

図 5.1: 信頼モデル

インターネットでは信頼モデルを確立するだけでなく、信頼した相手が本人であると確認できる機構が必要である。現在では、公開鍵暗号方式を利用した電子署名がよく用いられる。相手を認証するには、相手の公開鍵が正当なものであることを保証する仕組みが必要となる。この仕組みを公開鍵保証モデルと定義する。公開鍵保証モデルには以下の 2 つがある。

- 友人の輪
- 認証機関 (CA) による保証

友人の輪は、友人の公開鍵は正当なものであると信用し、その友人が署名する公開鍵は信用するモデルである。このモデルは PGP[9] で用いられている。このモデルは、当事者のみで公開鍵を保証でき、実現が容易であるという利点がある。しかし、人数が拡大すると処理できなくなる。

そこで、信頼された第三者機関によって公開鍵を保証するモデルが、インターネット上に構築された。このモデルは上位 CA が下位 CA の公開鍵を保証する。一番上に位置する CA をルート CA と呼び、階層的に公開鍵を保証する (図 5.2)。

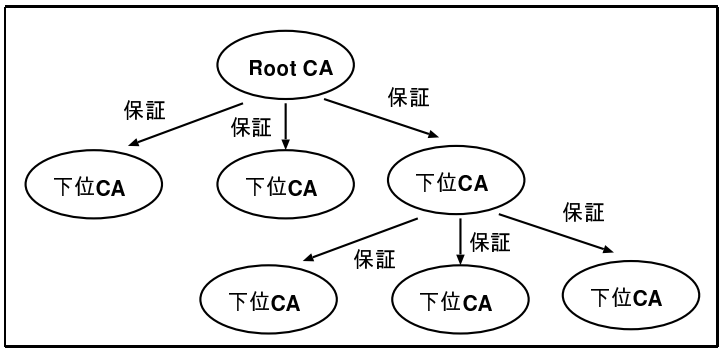

図 5.2: CA による公開鍵の保証

5.2 各情報収集形態における rics の安全性

この節では、インターネットを利用者を対象とした情報収集形態を分類し、情報収集形態に応じた環境の構築について述べる。そして、情報収集形態ごとに情報収集プロトコルの安全性に関して議論する。

情報収集形態に応じた環境の構築とは、まず情報収集者、対象者、集計サーバ、情報収集サーバを適切に配置することである。次に、情報収集形態に応じた信頼モデルを確立する必要がある。そして、公開鍵保証モデルによる公開鍵の保証の確立が必要である。この過程を経なければ、情報収集サーバや回答者が情報収集を拒否する可能性が高く、また安全性は保証できない。

情報収集形態を以下のように分類する。各情報収集形態については後で述べる。情報収集者は各情報収集形態で想定されている環境および目的で情報収集しなければならない。

- 情報収集形態 A
  - 情報収集形態 A-1
  - 情報収集形態 A-2
  - 情報収集形態 A-3
- 情報収集形態 B
  - 情報収集形態 B-1
  - 情報収集形態 B-2
  - 情報収集形態 B-3

情報収集形態 A は情報収集サーバのデータベースを使用して、データベースから対象者を抽出して情報収集する形態で、情報収集形態 B はデータベースに登録されている人以外の人を対象者として情報収集する形態である。

5.2.1 情報収集形態 A-1

情報収集形態 A-1 は、情報収集者は各サイトの対象者に対しアンケート調査し、情報収集者は自分のためのみに収集したデータを活用することを想定する。計算機環境は図 5.3 で示される。対象者は同一サイト内にある情報収集サーバのデータベースに登録されている。集計サーバは情報収集者によって起動されており、情報収集者は集計サーバを利用して各サイトの情報収集サーバと通信する。

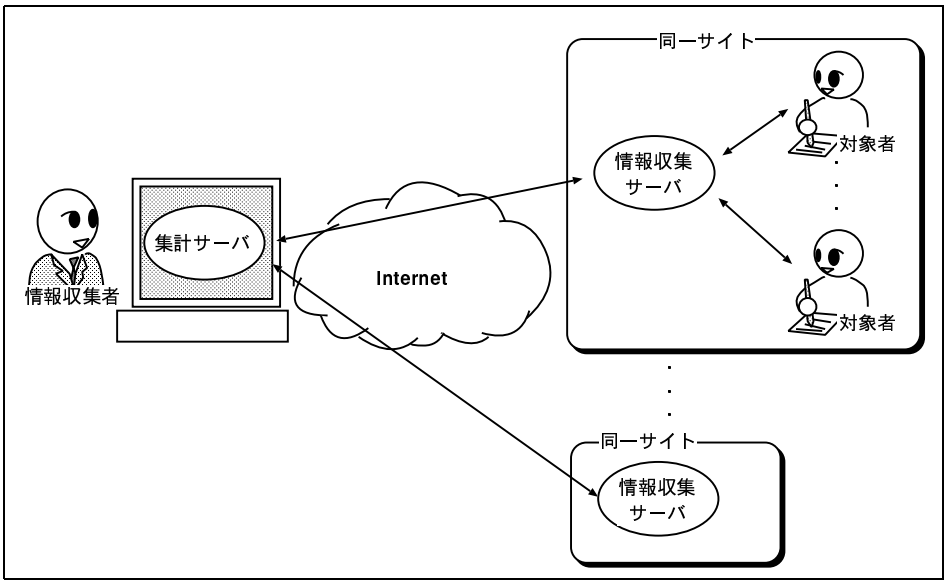

図 5.3: 情報収集形態 A-1

情報収集形態 A-1 では、対象者と情報収集サーバがお互いを信用し、情報収集サーバと集計サーバがお互いを信用する信頼モデルとなる (図 5.4)。対象者が情報収集サーバを信用できない場合は、データベースへの登録を拒否できる。情報収集サーバが対象者を信用できない場合は、対象者をデータベースに登録しなければよい。情報収集者が情報収集サーバを信用できない場合は、その情報収集サーバに情報収集を依頼しなければよい。情報収集サーバが集計サーバを信用できない場合は、その集計サーバからの接続を拒否できる。各公開鍵は友人の輪モデルにより保証する。

情報収集形態 A-1 で求められることは、情報収集者が正当な回答者から正当な回答データを取得できることである。そのためには、質問票と回答データの完全性を保持すること、回答者を認証すること、多重回答の防止が必要である。また、回答者のプライバシを保護するために、回答データの機密性を保持すること、匿名性の実現が必要である。

まず、機密性と完全性に関して議論する。情報収集形態 A-1 では、質問票が完全性を保持したまま対象者に送信され、回答データが機密性と完全性を保持したまま情報収集者に送信されることが要求される。情報収集クライアントと情報収集集サーバ間は同一計算機上で起動されているので、その間の通信の機密性と完全性は保持される。rics-send2 プロトコルにより、質問票定義コードは完全性を保持したまま集計サーバから情報収集サーバ

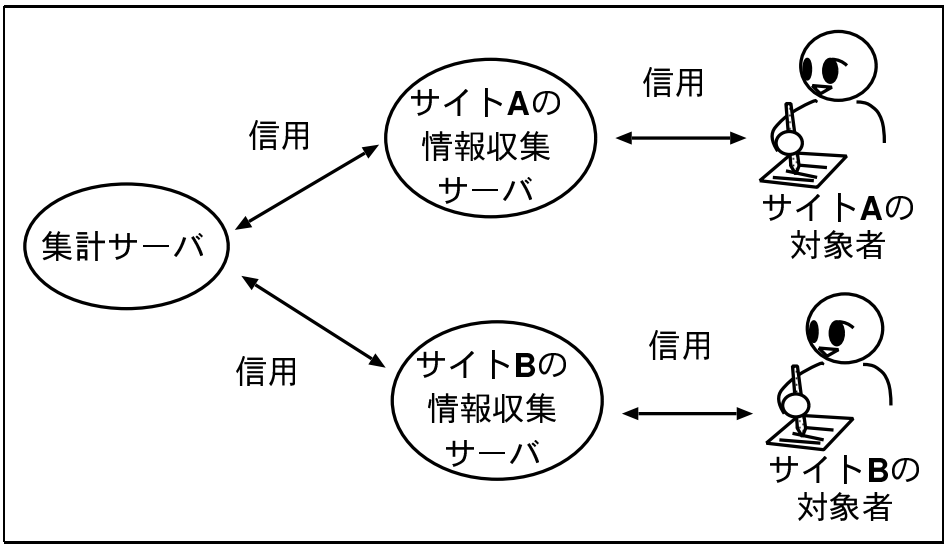

図 5.4: 情報収集形態 A-1 の信頼モデル

に送信できる。質問票の URL は、PGP を使用して電子メールで送信することにより、機密性と完全性を保持したまま対象者に送信できる。回答者は質問票および回答処理プログラムを情報収集サーバから HTTP を利用してダウンロードする。SSL を利用することにより、質問票と回答処理プログラムの完全性は保持できる。rics-collect1、rics-collect2プロトコルにより、回答データは機密性と完全性を保持したまま回答者から情報収集サーバを経由して集計サーバへ送信できる。ゆえに、情報収集形態 A-1 での情報収集では、機密性と完全性を保持できる。

次に認証に関して議論する。電子メールで質問票の URL を送信する段階では、質問票の URL を知り得るのは対象者のみである。しかし対象者が質問票を参照したときに、質問票の URL が洩れる可能性がある。ゆえに、第三者が質問票を参照することが可能となる。ゆえに認証は完全ではない。この問題点は、回答者が質問に回答する際に、秘密鍵の入力を求めることで解決できる。しかし、この解決方法は回答者に余計な作業を負担させ、回答率が低下することが予想される。

多重回答の防止は、ある URL からの回答を一回のみ受理することで実現できる。しかし、複数の情報収集サーバと通信する場合、別々のデータベースに同一回答者が登録されている可能性がある。回答者を認証するのは情報収集サーバであるため、異なる情報収集サーバに同一回答者が登録されている場合には、rics は同一回答者による多重回答を防止できない。しかし、インターネットでは、同一人物が容易に複数のアカウントを取得できるので、この問題を解決するのは困難である。この問題を解決するためには、インターネット上に個人を識別するための識別子を用意する必要があり、これは個人に背番号をつけることになり、また別の議論が必要となる。

匿名性は rics データフローモデルにより実現される。ただし、集計サーバと情報収集サーバが結託すれば、匿名性は保持されない。しかし、情報収集サーバと回答者は友人の輪によりお互いを信用しているので、情報収集サーバの管理者の不正はないと想定できる。

rics 以外にもサイト内ではさまざまなサービスが運営されており、管理者の不正を仮定すると、ほとんどのサービスを運営できない。結託の問題は、情報収集者と回答者の間に介在する機関を増やすことによって、全部の機関が結託する可能性を低くできる。しかし、この解決策は電子選挙に関する研究でも考えられているが、実現が困難である。

5.2.2 情報収集形態 A-2

情報収集形態 A-2 の計算機環境は情報収集形態 A-1 と同じである (図 5.3)。情報収集形態 A-2 は情報収集形態 A-1 とは目的が異なり、選挙を想定する。対象者は信頼できる第三者機関で動作している情報収集サーバのデータベースに登録されている。情報収集者も信頼できる第三者であり、情報収集者が集計サーバを起動する。

選挙の場合、情報収集サーバと集計サーバは全ての対象者から信用される必要がある。また、集計サーバと各情報収集サーバがお互いに信用し、情報収集サーバとそのデータベースに登録されている対象者がお互いに信用する必要がある。この信頼モデルでの公開鍵の保証は CA により保証するモデルとなる。ただし情報収集サーバとそのデータベースに登録されている対象者の間の信頼関係では、友人の輪モデルで公開鍵を保証する。

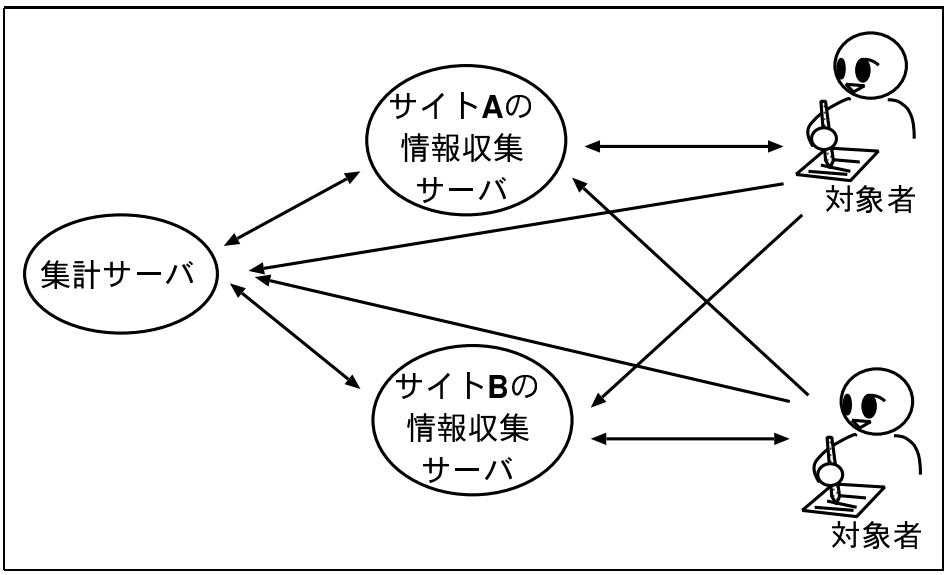

図 5.5: 情報収集形態 A-2 の信頼モデル

情報収集形態 A-2 では、情報収集形態 A-1 での要件がまず求められる。さらに、正当な集計結果を公開し、その集計結果を誰もが正当であると確認できる必要がある。

機密性、完全性、認証に関しての議論は情報収集形態 A-1 の場合と同様の議論ができる。

多重回答の防止は、ある URL からの回答を一回のみ受理することで実現できる。情報収集形態 A-2 の場合は、第三者機関が住民票などの何らかの方法で有権者を複数の情報収集サーバのデータベースに登録できないようにすることが求められる。

匿名性は rics データフローモデルにより実現される。情報収集形態 A-1 と異なり、集計サーバと情報収集サーバは信頼できる第三者機関により運営されるため、結託する状況は考えにくい。結託の問題に対して、情報収集者と対象者の間を中継する機関を増やす解決策があるが、この解決策は現実的でない。逆に、不正な回答があった場合に、結託することにより回答者を特定できる利点がある。この利点は電子決済で実際に利用されている。クレジットカード型の電子決済では、クレジットカード番号と購入した商品を保持する機関を分離することにより、匿名性を実現している。そして、不正な購入があった場合には、両者が結託することにより不正者を特定する。

選挙の場合は、情報収集者が集計結果を公開する必要がある。そのため、回答者が正当な集計結果であることを確認でき、正当な集計結果でない場合には異議申し立てができる機構が必要である。しかし、rics ではこの機構を提供していない。ゆえに、rics を選挙に利用するにはまだ解決すべき問題点があると言える。この機構の実現については、文献 [40][41] などで議論されている。

5.2.3 情報収集形態 A-3

情報収集形態 A-3 は集計サーバおよび情報収集サーバが同一サイト内に起動されている (図 5.6)。情報収集形態 A-3 は同一サイト内でのアンケート調査を想定している。情報収集サーバのデータベースに登録されている全ての人が、情報収集者および対象者になり得る。情報収集形態 A-1、情報収集形態 A-2 では、集計サーバを情報収集者が起動していたが、情報収集形態 A-3 の場合には、集計サーバはサイト内のある管理者によって運営されている。ただし、情報収集者が自分で集計サーバを起動させてもよい。

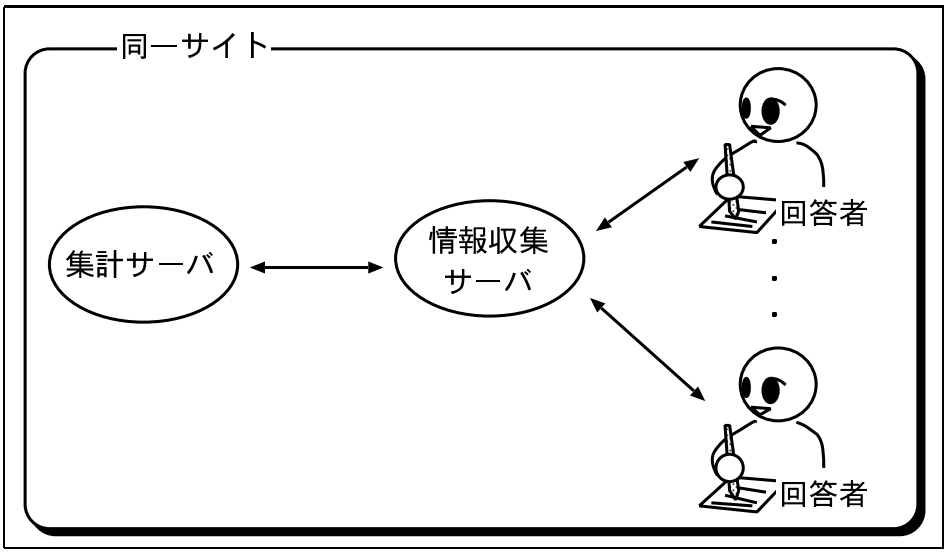

図 5.6: 情報収集形態 A-3

情報収集形態 A-3 では、集計サーバと情報収集サーバが異なる管理者によって管理され

る必要がある。信頼モデルは、集計サーバおよび情報収集サーバがデータベースに登録されている全ての人から信用されるモデルになる (図 5.7)。公開鍵の保証は、友人の輪モデルで実現できる。

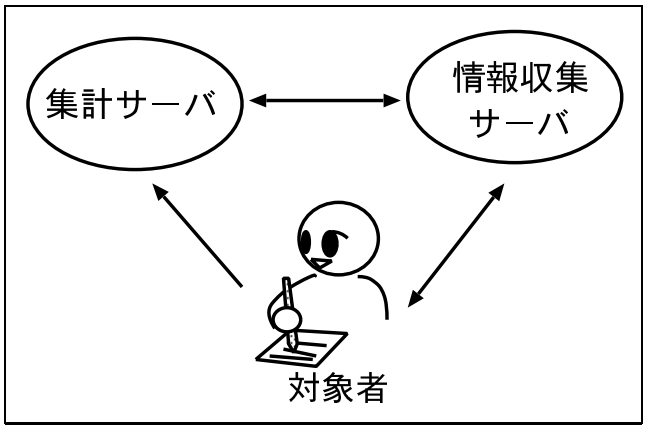

図 5.7: 情報収集形態 A-3 の信頼モデル

情報収集形態 A-3 では、情報収集形態 A-1 での要件が求められる。機密性、完全性、認証、匿名性に関しては情報収集形態 A-1 と同様の議論ができる。ただし、機密性について問題がある。情報収集形態 A-3 では、情報収集者と集計サーバがネットワーク的に異なる位置にいるため、情報収集者が集計結果を集計サーバからダウンロードする作業が伴い、その際の機密性は保持できない。集計結果の URL が洩れる可能性があるためである。よって集計結果が洩れても影響がない場合にのみ rics を利用できる。集計結果の機密性を保持したい場合は、情報収集者が集計サーバを起動すればよい。

多重回答は、データベースに同一人物を 2 重に登録しなければ防止できる。同一サイト内であれば、これは容易に実現できる。

同一サイト内では、rics による情報収集機構の構築も容易であり、rics を利用することで安全性の高い情報収集ができると考えられる。

5.2.4 情報収集形態 B-1

情報収集形態 B-1 は図 5.8 で示される計算機構成になる。情報収集者が集計サーバと情報収集サーバを起動する。情報収集形態 B-1 はインターネット上で不特定多数の人を対象とした情報収集を想定している。例えばインターネット上での学会やチケットの申込である。

情報収集形態 B-1 での信頼モデルは、回答者が情報収集サーバを信頼するモデルとなる。公開鍵の保証は、情報収集サーバの公開鍵が CA により保証されるモデルとなる。

情報収集形態 B-1 では、質問票の完全性、回答データの機密性と完全性の実現が求められる。SSL を利用することにより、質問票の完全性を実現できる。そして、rics-collect1 プロトコルにより回答データの機密性と完全性も実現できる。

情報収集形態 B-1 では rics 匿名性、認証、多重回答の防止を実現できない。回答者を認

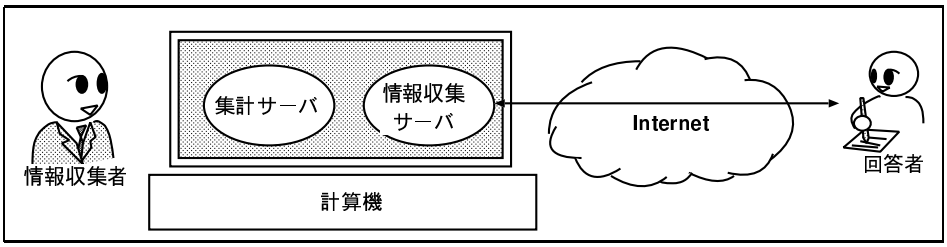

図 5.8: 情報収集形態 B-1

証して多重回答を防止するには、回答者に電子メールアドレスのような回答者情報を求める必要がある。ゆえに、情報収集形態 B-1 では学会の受け付けなど、回答者情報が必要な情報収集のみに利用できる。この場合には匿名性を実現する必要はない。

5.2.5 情報収集形態 B-2

情報収集形態 B-2 は図 5.9 で示される計算機構成になる。情報収集者が集計サーバと情報収集サーバを起動する。情報収集形態 B-2 は、オフライン上の不特定多数の人を対象としたアンケート調査を想定している。例えば街角アンケート調査である。街角に計算機を一台起き、計算機の前で情報収集者が回答者に対しアンケート調査する。

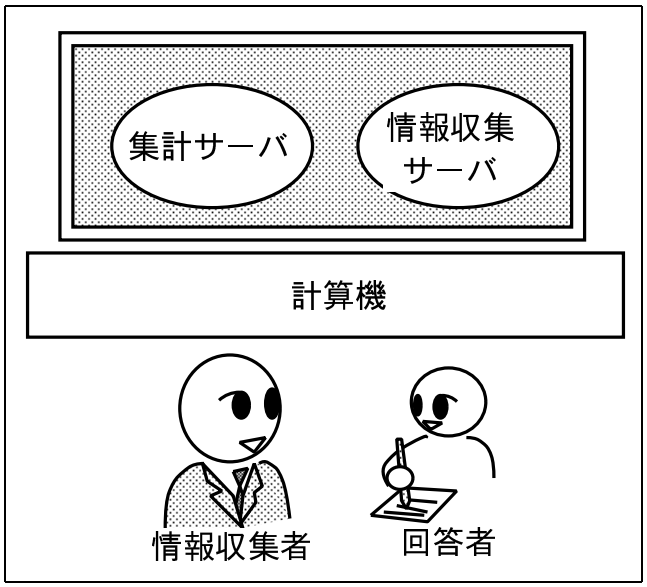

図 5.9: 情報収集形態 B-2

情報収集形態 B-2 はインターネットを利用しないので、機密性、完全性は保持できる。回答者の認証は情報収集者自身が行う。匿名性は情報収集者が対面している回答者を知らなければ実現できる。情報収集者と回答者がお互いに面識がある場合には、匿名性を保持

できない。これは現在の街角アンケート調査でも同様である。多重回答は情報収集者が同一人物に回答を要求しないことで防止できる。

5.2.6 情報収集形態 B-3

情報収集形態 B-3 も情報収集形態 B-2 と同じ計算機構成になる。情報収集形態 B-3 では情報収集者は人ではなく、アプリケーションを想定している。あるアプリケーションがユーザに何らかの入力を求める際のインターフェイス部分に rics を使用する。情報収集形態 B-3 では、情報収集プロトコルで挙げた要件を特に必要としない。

5.3 安全性の向上に関する議論

この節では、情報収集形態に関係なく一般的な議論をする。特に結託に対する問題を解決する方法について関連研究を参考にして議論する。

これまでの考察から、異なる機関が結託すると匿名性が保持されないという問題が発生した。この問題に対し、情報収集者と回答者の間を中継する機関を増やす方法が考えられている。MIX ネットでは、全てのの機関が結託しない限り匿名性は保持される。そのため、中継する機関を増やせば、全ての機関が結託する確率は下がる。さらに MIX ネットにマルチパーティプロトコルの概念を導入し、さらに安全性を向上させる方法も考えられている [42](図 5.10)。

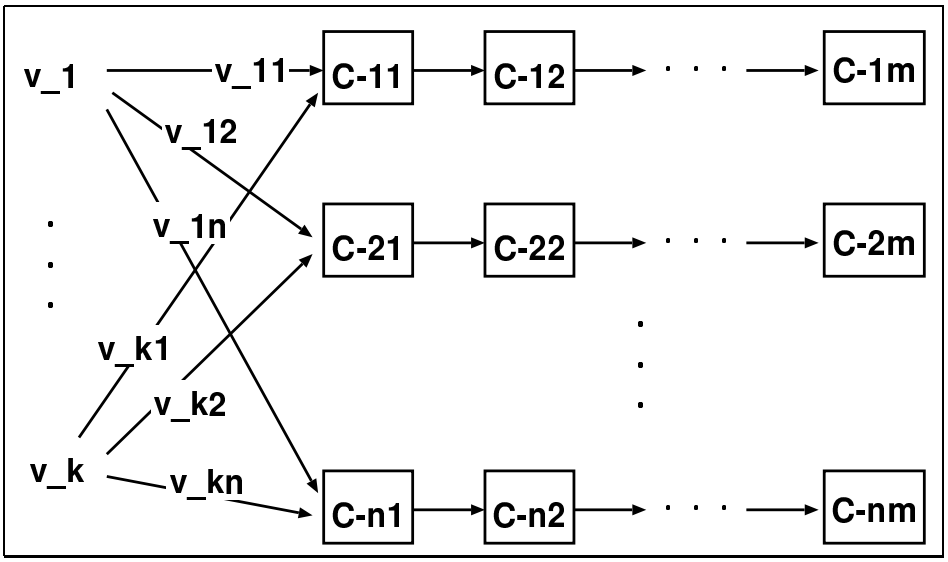

図 5.10: 安全性を向上させた MIX ネット

しかし、中継する機関を増やすと故障率が向上する。中継する機関の数を n、各機関の

故障率をαとすると、全体の故障率は以下のようになる。

$$1-(1-\alpha)^n \approx n\alpha$$

この問題の対し、半数未満の機関が故障しても回答データを復元できる方法が考案されている [25][43]。

5.4 まとめ

要求される安全性は情報収集形態ごとに異なる。この章では、情報収集形態ごとに実現できる安全性について評価・考察した。そして、適切に環境を構築することで、安全性の高い情報収集ができることを示した。

rics を利用するには、まず適切な信頼モデルを確立し、公開鍵の正当性が保証される必要がある。そのためには、rics の仕様を公開し、情報収集者と対象者に rics の動作原理を理解させる必要がある。rics がブラックボックスでは、ユーザは rics の動作を信用できない。

認証に関しては、その強度と利便性にはトレードオフの関係があり、rics では回答者の利便性を考慮して、回答時にパスワードの入力を求めなかった。回答時に回答者の秘密鍵を要求すれば、認証強度は向上する。また、rics は情報収集サーバと集計サーバの結託に弱い。結託する確率を下げるための方法については前節で述べた。結託を防ぐためにも、信頼モデルの確立が重要である。また結託した場合には、社会的制裁が加えられる制度も必要であると考えられる。

第6章　rics の利便性に関する評価と考察

この章では rics 全体としての評価と考察をする。特に rics アプリケーションが提供するユーザインターフェイス関して議論する。またスケーラビリティに関しても議論する。

6.1　WIDE ワークショップでの実験

この節では WIDE ワークショップでの実験について述べる。WIDE ワークショップは WIDE プロジェクト[1] のメンバーが集まって議論する場である。WIDE ワークショップは 1998 年 9 月 8 日〜11 日まで行なわれ、参加者は 235 人であった。WIDE ワークショップでは、仮設ネットワーク (図 6.1[2]) が用意され、ほとんどの人がインターネットに接続できる。筆者はこの WIDE ワークショップで、ネットワークの構築、運営に携わった。そして、このネットワークに rics を起動した。

6.1.1　実験概要

実験の目的は情報収集者および回答者の利便性の評価をすることである。また rics の動作確認である。

ワークショップでは、WWW 上にアンケート BOX を用意した。アンケート BOX は各参加者ごとに用意され、その URL を合宿参加者の名札の裏に書くことで、アンケート BOX の URL を全員に配布した。ゆえにアンケート BOX の URL を知り得るのは本人のみである。

この実験では、情報収集形態 A-3、情報収集形態 B-2 による情報収集機構を提供した。データベースにはアンケート BOX の URL が登録されている。情報収集者は電子メールで情報収集を依頼する。情報収集形態 A-3 の場合は、アンケート BOX にアンケート内容が追加される (図 6.2)。参加者全員のアンケート BOX に質問票が追加される。そして、集計結果の URL が電子メールで情報収集者に返信される。情報収集形態 B-2 の場合には、質問票が一つ作成され、集計結果の URL と質問票の URL が電子メールで情報収集者に返信される。

[1] http://www.wide.ad.jp
[2] WIDE プロジェクトによる作成

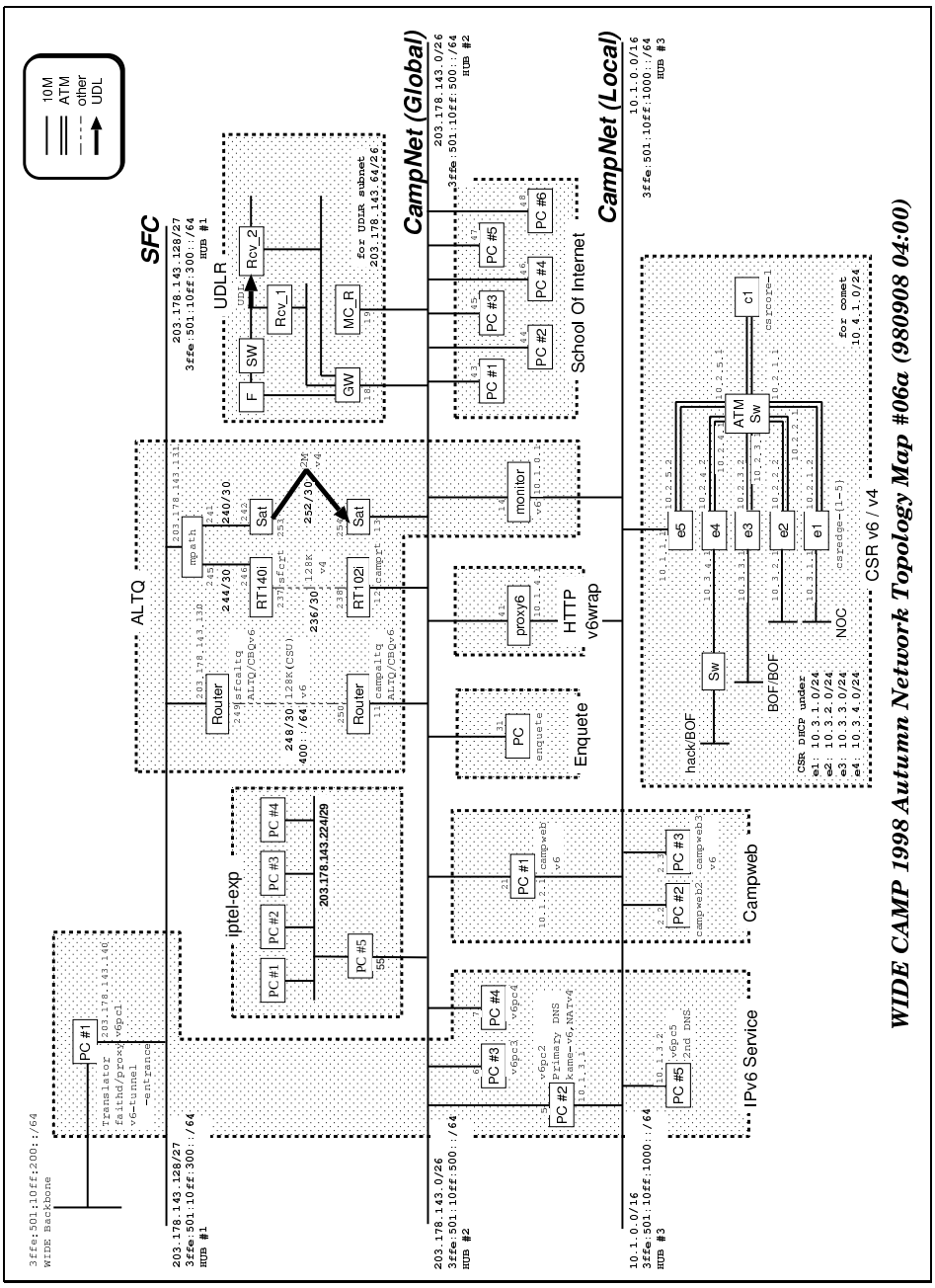

図 6.1: ワークショップにおける仮設ネットワーク

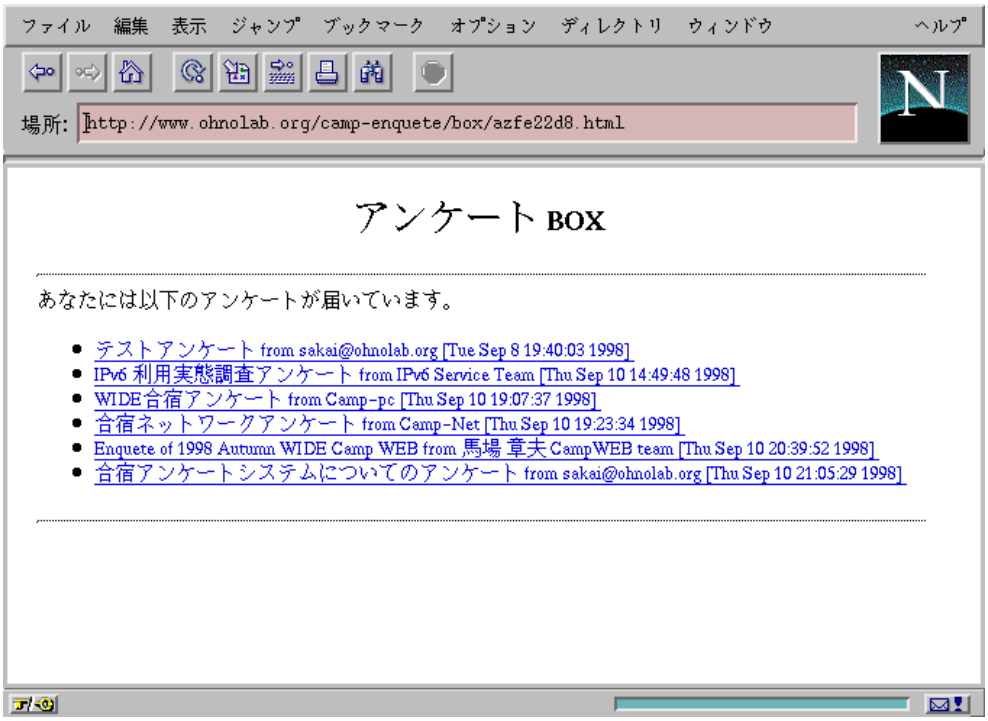

図 6.2: 各個人のアンケート BOX

6.1.2 実験結果

ワークショップ期間中、9 月 8 日 19:30〜9 月 11 日 10:20 まで rics を起動した。情報収集形態 A-3 では、表 6.1 に示される 6 つのタイトルのアンケート調査が実施された。回答者のアンケート BOX には図 6.2 のように表示される。表 6.2〜表 6.7 にその結果を示す。

情報収集形態 B-2 のアンケート調査は「IP 電話音質評価実験アンケート」というタイトルの 1 件のみ実施された。このアンケート調査は、ワークショップ会場のある場所に集まって、参加者に実験に参加してもらい、その場で一台の計算機に表示された質問票の質問に回答してもらった。また、情報収集形態 A-3、情報収集形態 B-2 での情報収集以外に、テストで rics 使用した件数が 38 回となった。どの情報収集者も実際のアンケート調査をする前に、質問票の見栄えや、どのように集計結果が作成されるのかなどを確認した。図 6.3 にワークショップ期間中に実施されたアンケート調査実施分布を示す (テスト使用を含む)。

次に、9 月 8 日 19:30 からの質問票への参照分布と回答の到着分布を図 6.4、図 6.5 に示す。質問票へのアクセスは全部で 975 件あり、のべ 703 件の回答があった。一時ネットワークが正常に運用されていなかったため、その期間の回答データ一件が、回収できなかった。それ以外の回答データは正常に回収できた。

アンケート名	タイトル
アンケート 1	テストアンケート
アンケート 2	IPv6 利用実態調査アンケート
アンケート 3	WIDE 合宿アンケート
アンケート 4	合宿ネットワークアンケート
アンケート 5	Enquete of 1998 Autumn WIDE Camp WEB
アンケート 6	合宿アンケートシステムについてのアンケート

表 6.1: 実施されたアンケート調査

アンケート実施開始時刻	1998/09/08/19:40:16
質問数	3
対象者数	235 人
質問票にアクセスした人数	111 人
アクセスしたが回答しなかった人数	40 人
回答者数	71 人
回答率	30 %

表 6.2: アンケート 1 の結果

アンケート実施開始時刻	1998/09/10/14:50:05
質問数	10
対象者数	235 人
質問票にアクセスした人数	130 人
アクセスしたが回答しなかった人数	9 人
回答者数	121 人
回答率	52 %

表 6.3: アンケート 2 の結果

アンケート実施開始時刻	1998/09/10/19:08:29
質問数	21
対象者数	235 人
質問票にアクセスした人数	138 人
アクセスしたが回答しなかった人数	8 人
回答者数	130 人
回答率	55 %

表 6.4: アンケート 3 の結果

アンケート実施開始時刻	1998/09/10/19:23:56
質問数	13
対象者数	235 人
質問票にアクセスした人数	125 人
アクセスしたが回答しなかった人数	7 人
回答者数	118 人
回答率	50 %

表 6.5: アンケート 4 の結果

アンケート実施開始時刻	1998/09/10/20:40:07
質問数	6
対象者数	235 人
質問票にアクセスした人数	109 人
アクセスしたが回答しなかった人数	10 人
回答者数	99 人
回答率	42 %

表 6.6: アンケート 5 の結果

アンケート実施開始時刻	1998/09/10/21:05:43
質問数	5
対象者数	235 人
質問票にアクセスした人数	102 人
アクセスしたが回答しなかった人数	9 人
回答者数	93 人
回答率	40 %

表 6.7: アンケート 6 の結果

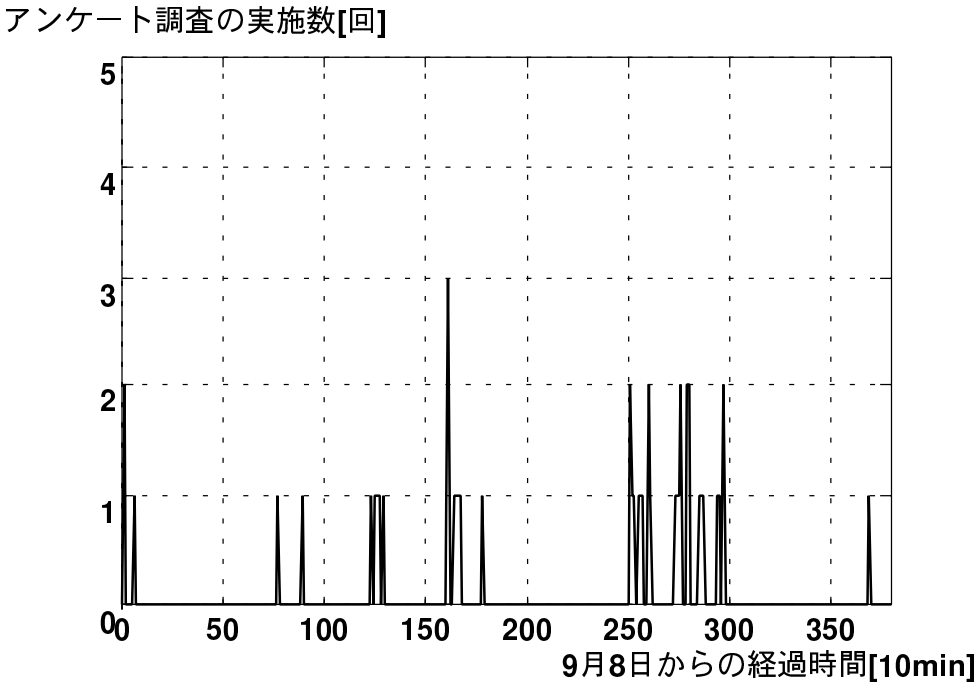

図 6.3: アンケート調査の実施分布 (10 分単位)

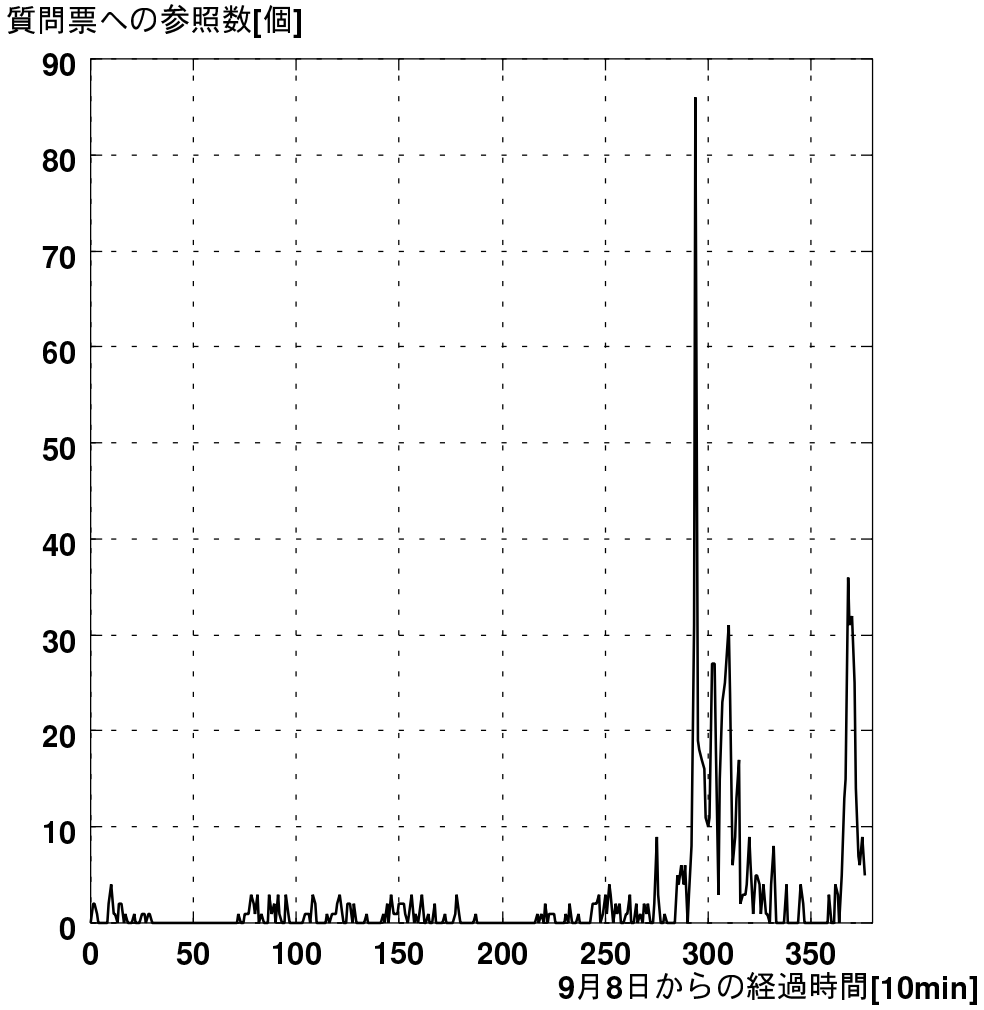

図 6.4: WIDE 合宿期間中における質問票への参照分布 (10 分単位)

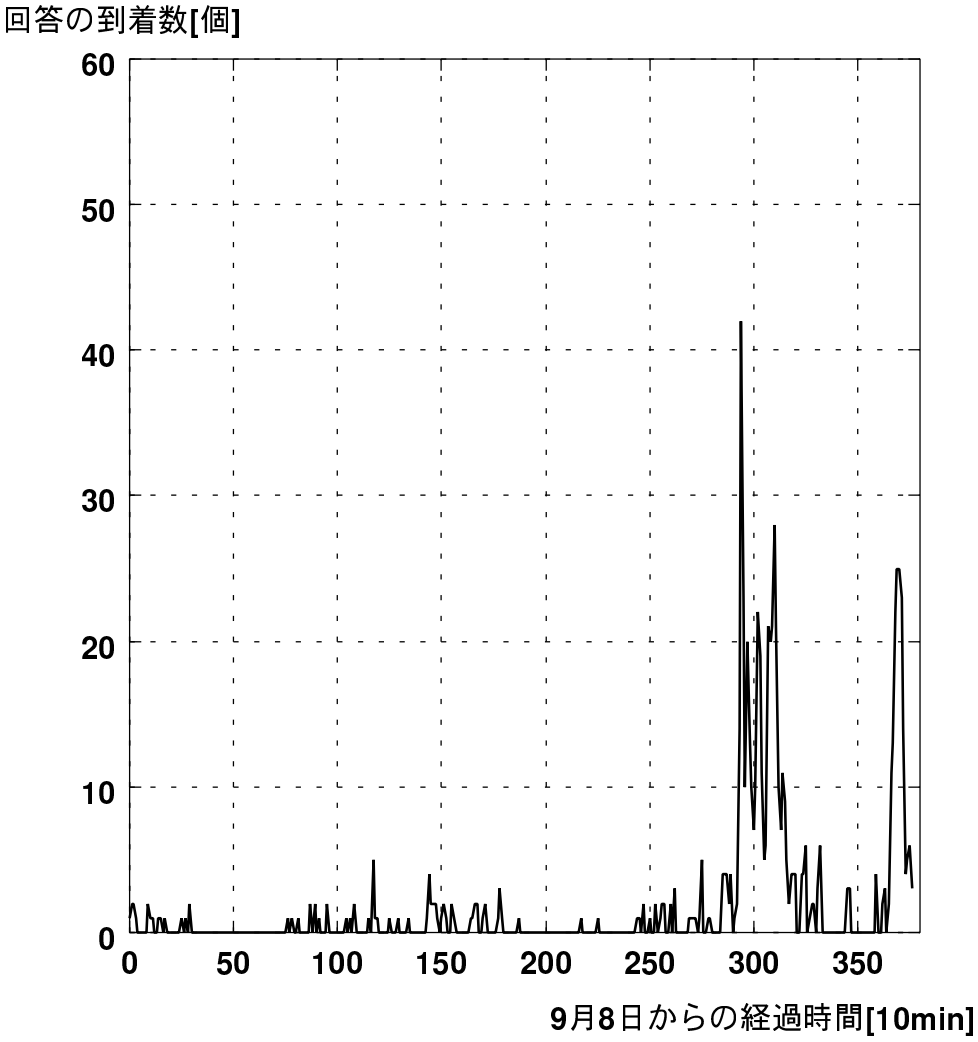

図 6.5: WIDE 合宿期間中における回答の到着分布 (10 分単位)

6.1.3 情報収集者の利便性に関する評価と考察

ワークショップ期間中の情報収集者は 6 人であった。情報収集者のうち 5 人はワークショップ期間中に rics の利用方法を覚え、アンケート調査を実施した。多くの情報収集者から、テキスト形式で質問内容を作成するのみで質問票が作成でき、自動的にアンケート調査が実施できることが評価された。

図 6.8 にワークショップ期間中に実施された主なアンケート調査の質問票の表示に必要な文字数の比較を示す。テキスト欄の数字は質問票の見栄えそのままをテキスト形式で記述した場合を数を示す。電子メールを利用して質問票を配布する場合には、テキスト形式で質問票をそのまま記述することが多い。WWW 質問票の欄には、同じ質問を HTML 形式で記述した場合の文字数が書かれている。WWW 質問票の場合にはさらに CGI を記述する必要があり、$+\alpha$ がそれを意味している。

表 6.8 から質問票定義コードの記述量は、質問票をテキスト形式でそのまま記述した場合の記述量と比較して、約 2 倍であることがわかる。WWW 質問票を作成する場合には、その記述量は、質問票をテキスト形式でそのまま記述した場合の記述量と比較して、5 倍以上という結果になった。

質問数	テキスト	質問票定義コード	WWW 質問票
3	143	325	$764 + \alpha$
6	498	868	$1875 + \alpha$
21	1471	2943	$8117 + \alpha$
13	675	1315	$2942 + \alpha$
9	444	959	$2194 + \alpha$
5	367	662	$1502 + \alpha$
22	1075	2146	$5053 + \alpha$

表 6.8: 文字数の比較

情報収集者のうちの 1 人は、当初、紙でのアンケート調査を計画していた。紙を利用する理由は、集計の手間を考慮すると CGI を利用したかったが、一から CGI を学習して質問票を作成するのが困難であり、紙を利用すれば簡単に質問票を作成できるためであった。この情報収集者に rics の使用感についてアンケート調査したところ、rics を利用することにより、紙の利用と比較して、アンケート調査の実施にかかる手間は約 2 割軽減し、集計にかかる手間は約半減したことがわかった。質問票定義コードの記述方法を覚えるのに少し作業量は増えるが、CGI よりも覚えやすく、慣れれば紙と比較しても手間はほとんど変わらないことがわかった。また、紙によるアンケート調査では、自分で質問票を配布する必要があるが、rics を利用すればその手間も省けた。集計の手間に関しては、紙の利用と比較して入力の手間が省けることが評価された。さらに、リアルタイムで集計結果がわかることも rics の利点として挙げられた。

しかし、質問票定義コードの記述方法が完全には理解できずにアンケート調査を実施で

質問	アンケートに回答する方法はすぐにわかりましたか？
回答者数	235 人中 93 人
すぐにわかった	77 人 (82.79%)
まあわかった	16 人 (17.20%)
あまりわからなかった	0 人 (0%)
全然わからなかった	0 人 (0%)

表 6.9: 回答方法についての質問

きなかった情報収集者もいた。アンケート調査が実施されたのは 45 件であったが、アンケート調査の依頼は 59 件あった。アンケート調査を実施できなかった理由は、すべて質問票定義コードの記述方法が間違っていたためである。どの情報収集者も質問票定義コードの記述を修正することでアンケート調査を実施できたが、1 人だけあきらめてしまった情報収集者がいた。その情報収集者からは、詳しいドキュメントとシステムの詳細を要求された。今後、ドキュメントの整備が必要である。また、質問票定義コードを記述するためのエディタも開発していきたい。

6.1.4 回答者の利便性に関する評価と考察

アンケート 6 での「アンケートに回答する方法はすぐにわかりましたか？」という質問では 8 割上の人が「すぐにわかった」と答えた (表 6.9)。また、何人かの回答者から回答者の匿名性を保証する仕組みがわからず不安であるという意見があった。ricsがブラックボックスではシステムを信用してもらうことができない。システムの動作についてのドキュメントを整備する必要がある。

6.1.5 その他の議論

最後に、その他実験結果からわかったことについて述べる。質問票への参照分布 (図 6.4)、回答データの到着分布 (図 6.5) から、千人規模の情報収集には十分耐えられることがわかる。

図 6.6 は質問票への参照時刻とその質問票に対する回答データの送信時刻の分布である。この分布は情報収集形態 A での情報収集のみに着目した。回答件数は 632 件であった。同じ質問票への参照が複数回ある場合はその参照時刻全てをプロットした。その場合の y 座標は最終的に回答データを送信した時刻になっている。すなわち、一度質問票を参照して、しばらく経ってから再び質問票を参照して回答した場合は、点の座標は $y > x$ かつ $y = x$ 直線から大きくはずれた領域にプロットされる。また、一度回答して再び質問票を参照した場合には、点の座標は $y < x$ の領域にプロットされる。複数回同じ質問票への参照があった件数は 118 件であった。$y < x$ の領域には 94 個の点があり、これは回答件数の 15 %にあたる。ゆえに多重回答防止の効果があったと言える。ほとんどの点は $y = x$ の近傍にあ

り、平均回答時間は 240 秒であった。複数回同じ質問票を参照した場合の回答時間は、回答したときの質問票参照時刻から回答データの送信時刻までとした。質問票を参照して、回答しなかった件数は 83 件であった。特にアンケート 1 でその件数が多い。これは、回答者が回答するアンケート調査を選んでいることを示している。

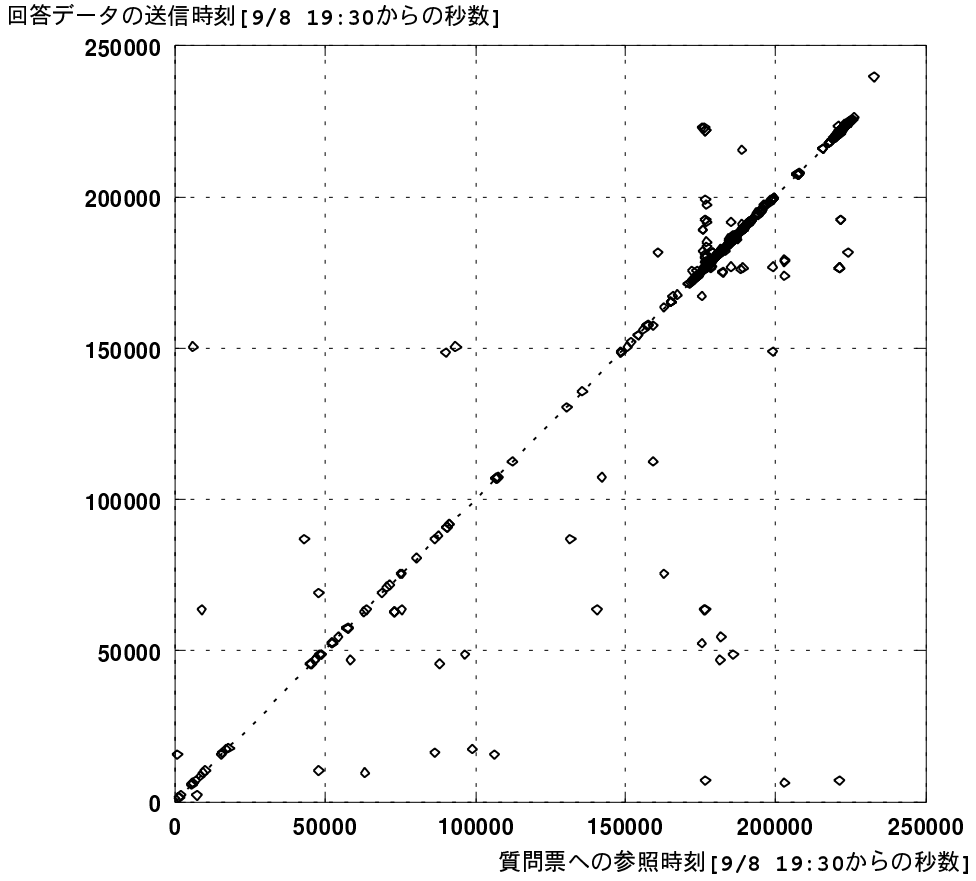

図 6.6: 質問票への参照時刻とその回答の到着時刻

6.2 大野研究室内での運用

1997 年 10 月より試作システムを大野研究室内で運用している。研究室内では、情報収集形態 A-3 の情報収集を提供し、電子メールを使用して質問票を配布している。データベースに登録されているのは、研究室に所属するメンバーの電子メールアドレスである。この運用では、情報収集対象グループを指定でき、研究室内メンバー全員、学生のみ、管理者などの指定ができる。情報収集者は必要な事項を書き、電子メールでアンケート調査を依頼する。その後、情報収集者は集計結果の URL を参照する。

以下では、rics の運用で実施された情報収集のうち 4 つの事例について述べる。

事例 1 1997 年 12 月 19 日から 1997 年 12 月 20 日まで、ある物品 B を研究室で購入してよ

いかという質問を、研究室内全員のメンバーに対して配布した。1 日で 15 人中 7 人からの回答があった。この結果が、その物品 B を購入するかしないかの決定、機種選定に役立った。

事例 2 1998 年 6 月 6 日から 1998 年 6 月 10 日まで、あるソフトウエア S が研究室内で使用されている数と、ソフトウエア S を研究室内で所有しているライセンス数について調査した。回答者には以下のことについて質問した。

- 自分が管理している計算機で、ソフトウエア S を使用している数
- 自分が管理している計算機のために購入したソフトウエア S の数

15 人中 11 人からの回答があった。その結果、ソフトウエア S に関してのライセンス数の過不足状況を把握できた。

事例 3 1998 年 7 月 17 日から 1998 年 7 月 20 日まで、研究室内で管理者権限所有者を新たに選出するための選挙を実施した。現管理者権限所有者に対して、 新管理者権限所有を希望する者を信任するかしないかの質問をした。 8 人中 5 人からの回答があった。しかしこの選挙では、目的や選出基準を決めておかなかったため、最終的には現 root 権限所有者間でのミーティングにより新管理者権限所有者を選出せざるをえなかった。

事例 4 1998 年 9 月 20 日から 1998 年 9 月 24 日まで、ミーティングの日程調整をするための調査をした。大野研メンバーに対し、ミーティングの開催希望日を質問した。15 人中 7 人のからの回答があり、ミーティングの日程調整に役立った。

以上のように、さまざま場面で rics が利用されている。上記に挙げた事例は、これまでは一人が手間をかけて調査する、またはミーティングを開催するなどの手段により、意思決定や問題解決がなされていた。上記に挙げた事例以外にも、研究室の物品を破損した場合の対処方法の決定にも利用された。また、学生同士で夕食に行く場所の決定など、学生同士のコミュニケーションとしても利用された。

6.3 プログラムのインターフェイスへの利用

この節では rics の情報収集形態 B-3 での利用について述べる。rics は集計データを標準入力としてコマンドを実行できる。この機能を利用すると、アプリケーションのインストール画面、設定画面など、プログラムのユーザインターフェイス部分に rics を利用できる。

例えば、ユーザに IP アドレスを設定してもらう場合を考える。図 6.7 のような質問票定義コードを記述する。

図 6.7 のファイル名を`config-ipaddr.xml` として、以下のコマンドを実行すると、図 6.8 のような質問票が表示される。

```
netscape ‘java ClientCom -f sakai -m b -d example.xml -h localhost -q‘ &
```

```
<?xml version="1.0" ?>
<!DOCTYPE questionnaire SYSTEM "questionnaire.dtd">
<questionnaire>
<title> PICKLES Configuration </title>

<item id="q1">
<free_answer>
<question> Enter IP address of this machine. </question>
<blank width="20" height="1"></blank>
</free_answer>
</item>

<exec> config-ipaddr </exec>
</questionnaire>
```

図 6.7: IP アドレスの設定画面の質問票定義コード

ユーザが設定画面の質問に回答すると、集計サーバにより集計データが作成され、質問票定義コードの`exec` タグに書かれている`config-ipaddr` プログラムを実行する (図 6.9)。`config-ipaddr` プログラムが実際に IP アドレスの設定をするプログラムである。

このように、質問票の作成から集計データの作成までを rics に頼れば、プログラムからユーザインターフェイス部分を切り離せる。そして容易にインターフェイス部分を作成できる。rics を利用すれば BSD/OS に添付されている`maxim` のような環境設定支援プログラムの作成に要する作業量はかなり軽減される。

6.4 スケーラビリティ関する評価と考察

rics は最初は C 言語を用いて実装された。回答処理プログラムは perl 言語で記述され、CGI として実装されていた。そのため作成できる質問票は制限されていた。ワークショップでの実験は C 言語での実装である。その後、システムの拡張、特にセキュリティの向上と質問票の拡張を目的に、rics は JAVA 言語で書き直された。これにより処理速度は下がった。

以下では、C 言語での実装、JAVA 言語での実装においての処理速度およびディスク使用量について述べる。実験環境は表 6.10 の通りである。

図 6.10 は、C 言語、JAVA 言語での実装で、質問票に記載される質問数を変化させた場合の処理時間およびディスク利用量の変化である。処理時間は、情報収集クライアントが集計サーバに接続した時刻から、集計サーバが情報収集クライアントの要求を処理し終えた時刻までである。C 言語と JAVA 言語での縦軸の違いは、対象者数が異なるため比較できない。

図 6.10 より、処理時間は、C 言語での実装は質問の数に依存せず、JAVA 言語での実装

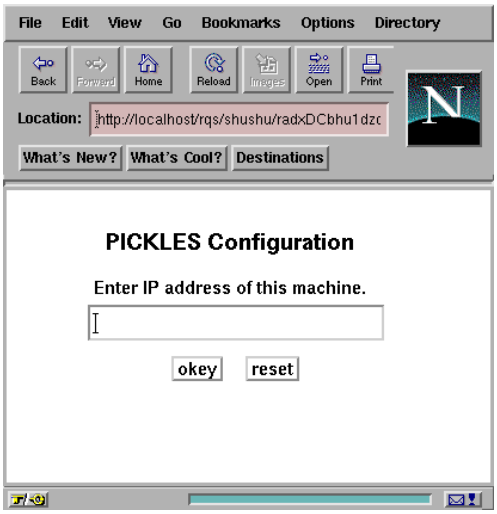

図 6.8: 設定画面

```
> config-ipaddr <<EOF
<?xml version="1.0" ?>
<!DOCTYPE answers SYSTEM "answers.dtd">
<answers>
<answer>
<item id="q1">
<free_answer>
<answer_text>131.112.57.33</answer_text>
</free_answer>
</item>
</answer>
</answers>
EOF
```

図 6.9: 集計データを引数としたコマンドの実行

OS	FreeBSD2.2.7
CPU	MMX Pentium 200MHz
Memory	96MB

表 6.10: 実験環境

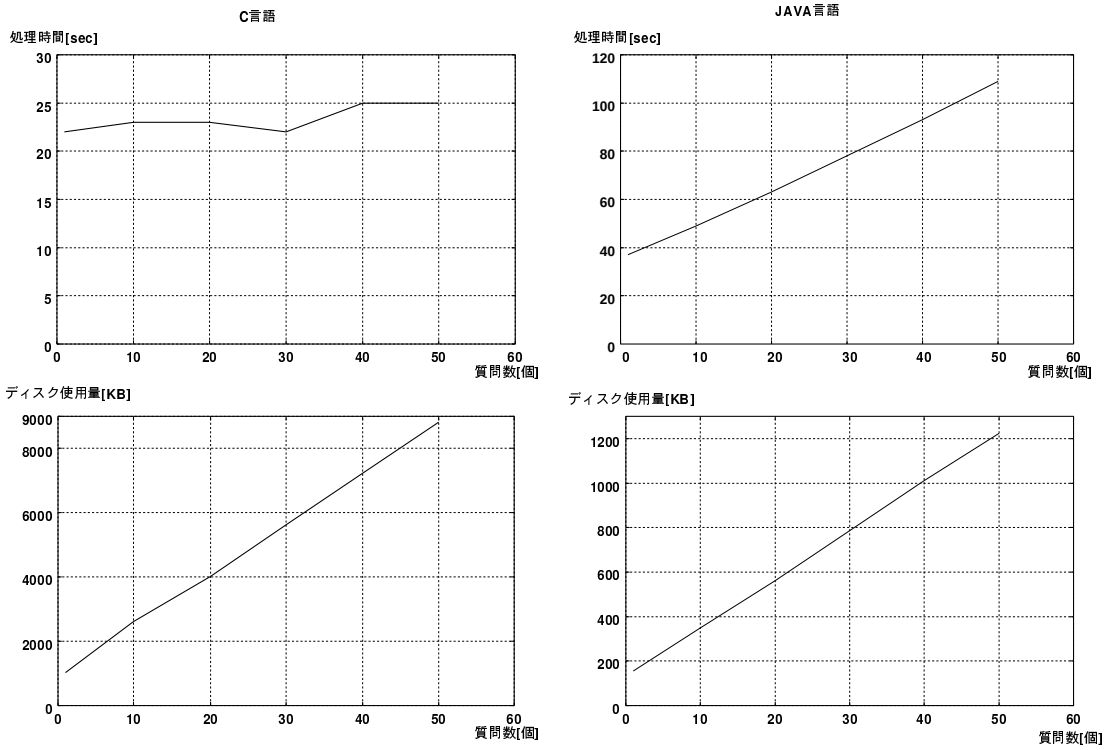

図 6.10: 質問数を変化させた場合

は線形に増加することがわかる。ディスク使用量はどちらも実装も線形に増加することがわかる。

図 6.11 は、C 言語、JAVA 言語での実装で、対象者数を変化させた場合の処理時間およびディスク利用量の変化である。処理時間は、情報収集クライアントが集計サーバに接続した時刻から、集計サーバが情報収集クライアントの要求を処理し終えた時刻までである。C 言語と JAVA 言語での縦軸の違いは、質問数が異なるため比較できない。

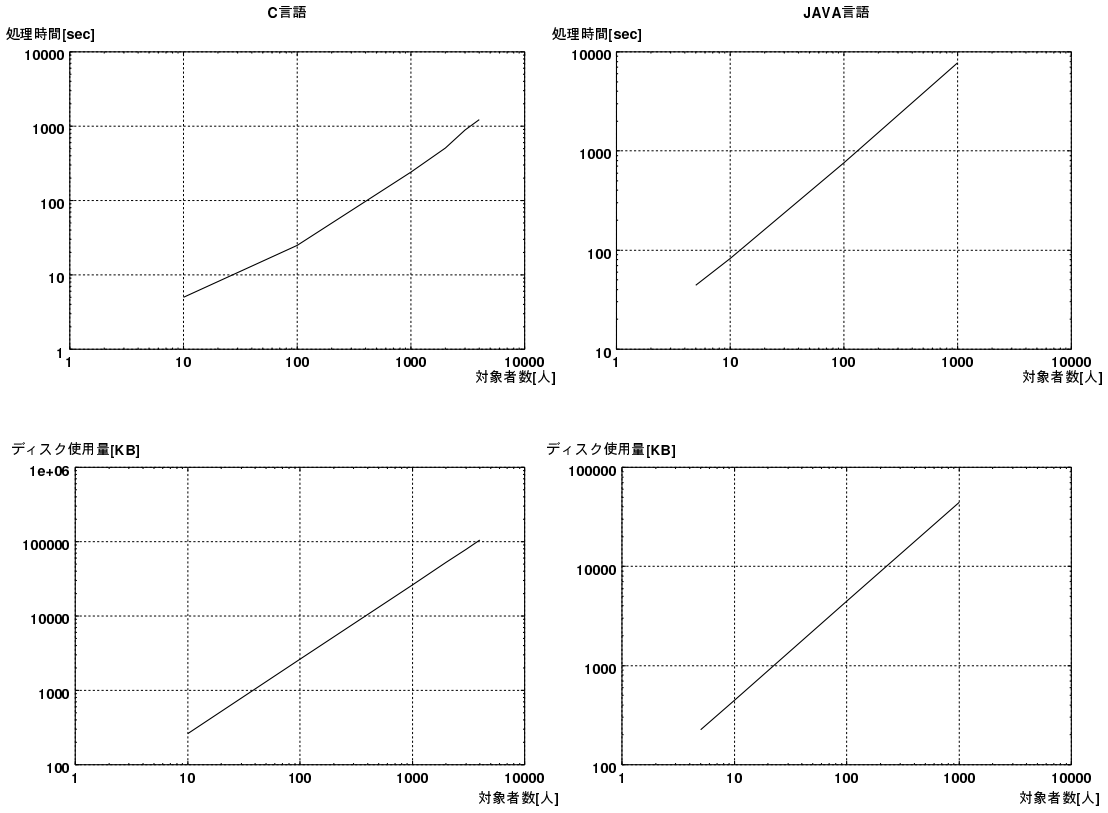

図 6.11: 対象者数を変化させた場合

図 6.11 より、対象者数が増加すれば、処理時間、ディスク使用量とも線形に増加する。JAVA 言語での実装では class ファイルを一つ作成するのに約 7 秒かかる。

以上の考察から、対象者数の増加、質問数の増加により、処理時間、ディスク使用量は増えることがわかる。rics は、負荷分散をするための機構を提供していない。ゆえに、一台の計算機では処理できる質問数、対象者数は制限される。しかし、増加関数が線形であるので、計算機の処理能力が向上すれば、処理できる質問数、対象者数も増加すると考えられる。回答データの到着の処理に関しては、到着間隔が大きいので、その対象者人数分の質問票が作成できれば処理できると考えている。今回の実験環境では、ワークショップの実験も示したように、千人規模の情報収集には十分に耐えられる。

しかし、スケーラビリティに関しての問題は対象者数ではなく、対象者をどのようにして集めるかである。ワークショップや大野研究室での実験で示したように、rics は情報収

集形態 A-3 では有効に機能した。対象者を容易に集めることができ、容易にデータベースを作成できるためである。

情報収集形態 A-1 のように、広域インターネット上において一般の人々を情報収集の対象としてアンケート調査する場合には、まず情報収集の依頼を受けてくれる情報収集サーバを見つけなくてはならない。そのためにも、rics を普及させ、インターネット上に存在する情報収集代行機関やプロバイダにも情報収集サーバが起動してもらう必要がある。

6.5 まとめ

rics を利用して情報収集した場合の作業量を、現在の情報収集の典型的な方法である、電子メールを利用して質問票を配布および回収する方法と、WWW 上に質問票を公開して質問票の URL を電子メールで送信する方法の 2 つと比較する。

質問票の作成にかかる手間は、6.1 節での実験で示したように、電子メールを利用する場合に比べて、rics は約 2 倍の記述量を要する。しかし、WWW を利用する場合と比較すると記述量は約半分である。次に、回答データを解析する手間について考える。電子メールを利用する場合は、到着した回答データが一定の形式になっていることは少なく、まず統一した形式にした回答データを作成する必要がある。一方、rics および WWW を利用した場合は、一定の形式で回答データを回収できる。ゆえに、以上の 3 つの方法の作業を比較すると表 6.11 のようにまとめられる。これより、rics が他の方法と比較して、情報収集に要する作業量を軽減していると言える。

質問票の作成	電子メールの利用 $< rics \ll WWW$ の利用
回答データの解析	WWW の利用 $\approx rics \ll$ 電子メールの利用

表 6.11: 作業量の比較

rics は情報収集形態 A-1、情報収集形態 A-3 のように、情報収集サーバを用意すれば、誰もが情報収集できるのが一つの特徴である。これまでは、人々を対象とした情報収集システムを作成したとしても、その作成者しかそのシステムを利用できなかった。また、そのシステムは質問の異なる情報収集には利用できなかった。慶応大学のインターネット授業調査アンケートシステム [2] は、インターネット上で高いセキュリティを持つ授業調査アンケートを実施できるが、誰もがこのシステムを利用してアンケート調査を実施すること、あらかじめ用意されている質問以外の質問を作成することは困難である。近年の WIDE ワークショップにおいても WWW を利用してアンケート調査が実施されていたが、その一回のアンケート調査のためだけにシステムを構築していた。rics を利用することで、誰もが容易に情報収集できるようになり、WIDE ワークショップにおいても複数のアンケート調査が実施された。

さらに rics は、現状の方法よりも多くの質問票を作成できることが特徴である。アナログ回答形式は紙を利用した情報収集では作成できなかった。

第7章　今後の展望

この章では、残された課題および rics の今後の発展について述べる。

7.1　CITRUS との連携

CITRUS は大学内での情報サービスを提供する。rics を CITRUS と連携して利用することにより、大学内での授業調査アンケート、質問票を使った履修登録などができる。今後 CITRUS の上に rics アプリケーションを実装していく。また、PICKLES プロジェクトでは ChipCard に代わる超小型携帯端末が考えられており、それも考慮する必要がある。

7.2　大規模な実験と運用

現在では、情報収集形態 A-1、情報収集形態 A-2、情報収集形態 B-1 での利用までに至っていない。複数の情報収集サーバを利用した情報収集、CA による公開鍵の保証モデルを必要とする情報収集である。そのためには、rics の普及、大規模な情報収集のための環境構築が必要である。

情報収集形態 A-1 では、アンケート調査を想定しているので、対象者全員に質問票を配布しなくても民意を反映したデータを取得できる。インターネットのトラフィックを軽減するためにも、データベースに登録されている対象者を適切にサンプリングする必要がある。rics では、回答データを逐次解析 [44] できるため、今までよりも少ないサンプルサイズで正確なデータを得られる。よって、今までとは違ったサンプリング方法が考えられる。一人ずつ質問票を配布して回答データを回収する方法がもっとも少ないサンプルサイズで正確なデータを得られるが、これでは時間がかかりすぎる。例えば、10 人ずつ抽出していく方法や、10 人、100 人、1000 人と抽出していく方法などが考えられ (図 7.1)。よって、どの時点で何人の対象者をサンプリングし、どの時点でサンプリングを終了するかを決定するアルゴリズムが必要となる。そして、情報収集期間、サンプルサイズ、データの信頼性を考慮した最適なアルゴリズムを考案していく必要がある。

7.3　質問票定義コードの拡張

rics は紙では実現できなかった連続量で回答できるアナログ回答形式を用意した。今後さらに、新しい質問形式を開発していく予定である。例えば、画像の利用 [4] などがある。

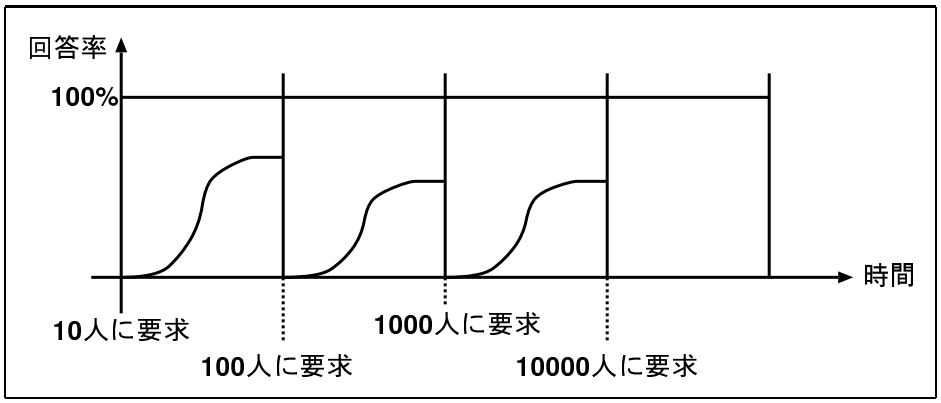

図 7.1: サンプリングの例

そのためには、どのような質問形式が回答者の考えをもっとも反映した回答を得られるかを考える必要がある。

また、現在では、回答によって質問の順番や質問の内容を変えるなどの、質問表示を制御するための機構が用意されていない。質問票定義コードに制御構造を導入する必要がある。

7.4 追加情報収集

インターネットを利用すれば、ある回答者に対して追加質問を容易にできる。現在の rics での情報収集は一往復のみを想定している。今後、回答者のプライバシを保護しつつ、追加質問できるような機構を構築していく。例えば、回答者 ID を用意することを考えている。回答者 ID から回答者を対応づけられるのは情報収集サーバのみであり、集計サーバは回答者 ID と回答データを保持する。このようにすれば、集計サーバが追加質問したい回答者 ID を選び、情報収集サーバがその回答者 ID を持つ回答者に追加質問できる。

7.5 配布環境の整備

rics を利用して情報収集した人はいるが、集計サーバ、情報収集サーバを筆者以外で起動した人はまだいない。WIDE ワークショップでのアンケート調査では rics を利用してみたい人がいることがわかった (表 7.1)。また、WIDE プロジェクト内では実際に rics を利用してみたいという声も上がっている。そのために、rics を配布できるようにパッケージ化する必要がある。また、ドキュメントの整備も必要である。

質問	本システムを自分のところでも利用してみたいと思いますか？
回答者数	235 人中 83 人
とても利用してみたい	17 人 (20%)
まあ利用してみたい	35 人 (42%)
どちらともいえない	27 人 (33%)
あまり利用したくない	3 人 (4%)
全然利用したくない	1 人 (1%)

表 7.1: 本システムを利用してみたいかについての質問

第8章　おわりに

本論文では、インターネット利用者を対象として情報収集するための機構rics を提案した。rics は階層構造を持ち、それぞれの階層ごとに役割を規定した。rics プロトコルは機密性と完全性を実現し、rics データフローモデルは匿名性、認証、多重回答の防止を実現した。rics アプリケーションは情報収集者と回答者へのユーザインターフェイスを提供した。

そして、rics プロトコルおよび rics データフローモデルにより、インターネット上に信頼性の高いデータを取得できる情報収集プロトコルを確立できた。rics アプリケーションにより、情報収集者および回答者の利便性が向上し、迅速な情報収集につながった。特に、質問票定義コードおよび集計データを標準化し、既存の情報交換システムに依存しないようにしたことにより、汎用的な質問票が作成できるようになり、質問票や集計結果の作成が容易になった。

rics は想定された情報収集形態のみで使用できる。実験および運用から、rics はアンケート調査、意志決定、スケジュール調整、アプリケーションの設定画面など、さまざまな用途に利用できた。今後は、まだ運用段階に至っていない大規模化を中心に開発を進めていきたい。

謝辞

3 年間熱心に指導して下さった大野浩之講師に感謝します。研究を進めるにあたり、多くの指導をして下さった郵政省通信総合研究所の中川晋一氏に感謝します。電子選挙に関して多くの時間を割いて説明して頂いた東京工業大学の黒沢馨教授に感謝します。貴重な議論の場と多くのアドバイスを頂いた WIDE プロジェクトのメンバーに感謝します。研究生活をさまざま形で支えてくれた大野研究室のメンバーに感謝します。

参考文献

[1] 日本インターネット協会. インターネット白書'98. 株式会社インプレス, 1998.

[2] 伊集院百合, 大川恵子, 村井純. インターネット授業調査アンケートシステムの設計と実装. インターネットコンファレンス'98 論文集, pp. 31–39. 日本インターネット協会, 日本ソフトウエア科学会インターネットテクノロジ研究会, 日本 UNIX ユーザ会, WIDE プロジェクト, December 1998.

[3] 浜砂敬郎. 統計調査環境の実証的研究. 産業統計研究社, 1990.

[4] 澤野貴, 酒井順一, 高橋俊二, 常澤邦幸, 羽生田浩教, 伊輿田光宏. WWW を利用したイメージアンケートシステム. 情報処理学会第 55 回 (平成 9 年後期) 全国大会講演論文集 (分冊 4), p. 331. 情報処理学会, September 1997.

[5] 酒井淳一, 大野浩之. インターネットを利用した安全かつ効率的なアンケート調査. 第 24 回グループウェア研究会研究報告. 情報処理学会, September 1997.

[6] 酒井淳一, 大野浩之. 小規模な組織の運営を支える情報共有機構-(3) アンケート調査による情報収集-. 第 55 回全国大会講演論文集 (分冊 4), pp. 341–342. 情報処理学会, September 1997.

[7] Alan O. Freier, Philip Karlton, and Paul C. Kocher. The SSL Protocol Version 3.0. INTERNET-DRAFT, November 1996.

[8] *SSH Protocols and Secure Shell.* http://www.ssh.fi/sshprotocols2/index.html.

[9] Simson Garfinkel. *PGP:Pretty Good Privacy.* O'Reilly & Associates, 1994.

[10] J. Linn. Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures. Request for Comments: 1421, February 1993.

[11] S. Kent. Privacy Enhancement for Internet Electronic Mail: Part II:Certificate-Based Key Management. Request for Comments: 1422, February 1993.

[12] D. Balenson. Privacy Enhancement for Internet Electronic Mail: Part III:Algorithms, Modes, and Identifiers. Request for Comments: 1423, February 1993.

[13] B. Kaliski. Privacy Enhancement for Internet Electronic Mail: Part IV:Key Certification and Related Services. Request for Comments: 1424, February 1993.

[14] Worldtalk Blake Ramsdell. S/MIME Version 3 Certificate Handling. INTERNET-DRAFT, December 1998.

[15] Worldtalk Blake Ramsdell. S/MIME Version 3 Message Specification. INTERNET-DRAFT, December 1998.

[16] *anonymizer*. http://www.anonymizer.com.

[17] Paul F. Syverson, David M. Goldschlag, and Mechael G. Reed. Anonymous Connections and Onion Routing. In *IEEE Symposium on Security and Privacy*, pp. 44–54, 1997.

[18] David M. Goldschlag, Mechael G. Reed, and Paul F. Syverson. Hiding Routing Information. In *Workshop on Information Hiding*, 1996.

[19] Mechael G. Reed, Paul F. Syverson, and David M. Goldschlag. Proxies for Anonymous Routing. In *12th Annual Computer Security Applications Conference*, 1996.

[20] Michael K. Reiter and Aviel D. Rubin. Crowds: Anonymity for Web Transactions. Technical report 97-15, DIAMACS, 1997.

[21] David Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, Vol. 24, No. 2, pp. 84–88, February 1981.

[22] 岡本龍明, 山本博資. 現代暗号. 産業図書株式会社, 1997.

[23] 藤崎英一郎, 太田和夫, 岡本龍明. 投票所を仮定した実用的な電子投票方式. 電子情報通信学会技術報告書, ISEC93-24. 電子情報通信学会, August 1993.

[24] 小山謙二. RSA 公開鍵暗号を用いた安全な無記名投票方式. 電子通信学会論文誌, Vol. J68-D, No. 11, pp. 1956–1966, November 1985.

[25] Wakaha Ogata, Kaoru Kurosawa, Kazue Sako, and Kazunori Takatani. Fault Tolerant Anonymous Channel. In *Information and Communications Security(ICICS'97)*, Vol. 1334 of *Lecture Note in Computer Science*, pp. 440–444, 1997.

[26] Josh Cohen Benaloh and Moti Yung. Distributing the Power of a Gevernment to Enhance the Privacy of Voters. In *Proceedings of the fifth Annal ACM Symposium on Principles of Distributed Computing*, pp. 52–62, August 1986.

[27] Josh D. Cohen and Michael J. Fischer. A Robust and Verifiable Cryptographically Secure Election Scheme. In *IEEE 26th Annual Symposium on Foundation of Computer Science*, pp. 372–382, Octber 1985.

[28] 岡本龍明, 大田和夫. 暗号・ゼロ知識証明・数論. 共立出版株式会社, 1995.

[29] David Chaum. Security without Indentification: Transaction systems to Make Big Brother Obsolete. *Communications of the ACM*, Vol. 28, No. 10, pp. 1030–1044, October 1985.

[30] 太田和夫. 単一の選挙管理者を用いた電子投票方式. 昭和 63 年電子情報通信学会春季全国大会, p. 296. 電子情報通信学会, 1988.

[31] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A Practical Secret Voting Scheme for Large Scale Elections. In *Auscrypt'92*, Lecture Note in Computer Science, pp. 244–251. Springer-Verlag, 1992.

[32] 岡本龍明. 暗号と情報セキュリティ. 日経 BP 社, 1998.

[33] *SET Specification*. http://www.visa.com/cgi-bin/vee/nt/ecomm/set/intro.html.

[34] Tim Bray, Jean Paoli, and C. M. Sperberg-McQueen. Extensible Markup Language (XML) 1.0. http://www.w3.org/TR/REC-xml, February 1998.

[35] 木本雅彦, 大野浩之. 学内情報システム ～CITRUS の概要～(大会優秀賞受賞). 情報処理学会第 52 回 (平成 8 年前期) 全国大会講演論文集 (1), pp. 277–278. 情報処理学会, March 1996.

[36] 木本雅彦, 大野浩之. 小規模な組織の運営を支える情報共有機構 (4) - 情報登録・抽出インタフェース -. 情報処理学会第 57 回 (平成 9 年後期) 全国大会講演論文集 (1), pp. 343–344. 情報処理学会, September 1997.

[37] 二ノ宮寿之, 木本雅彦, 大野浩之. 公衆情報端末と超小型携帯端末の連携によるアンケート調査システムの構築. 情報処理学会第 57 回 (平成 10 年後期) 全国大会講演論文集 (1), pp. 581–582. 情報処理学会, September 1998.

[38] Masahiko Kimoto and Hiroyuki OHNO. A way to the ubiquitous computing: Design and implementation of the pickles information kiosk. In *Proceedings of The International Workshop on Asia-Paciffic area advanced research information sharing technology*, pp. 67–74, March 1998.

[39] 木本雅彦, 大野浩之. 自律型ネットワーク端末 (pickles) を用いたシステム運用技法. 分散システム運用技術シンポジウム'98 論文集, pp. 93–99. 情報処理学会, February 1998.

[40] 佐古和恵. 異議申し立てを考慮した電子投票システム. 電子情報通信学会技術報告書, ISEC92-56. 電子情報通信学会, 1992.

[41] 佐古和恵, Joe Kilian. 全体検証可能な電子投票方式. 暗号と情報セキュリティシンポジウム, SCIS95-B1.3. 電子情報通信学会, January 1995.

[42] 秋山稔, 田中良明, 新保淳. 多重暗号化無記名投票方式. 電子通信学会論文誌, Vol. J69-B, No. 4, pp. 314–323, April 1986.

[43] 尾形わかは, 黒沢馨, 高谷和伯. (k,n) 閾値匿名通信路とその応用. 暗号と情報セキュリティシンポジウム, SCIS97-27F. 電子情報通信学会, February 1997.

[44] P.G. ホエール. 入門数理統計学. 培風館, 1978.

付 録A　質問票定義コードの詳細

現在、質問票定義コードにはtitle タグ、item タグ、text タグ、exec タグが定義されている。title タグには情報収集のタイトルを記述する。item タグには一つの質問を記述する。item タグには id 属性が必須である。id 属性は一つの質問とそれに対する回答を対応づけるために利用される。text タグは質問票に質問以外の文章を表示するために使用し、text タグには実際に質問票に表示する文章を記述する。exec タグには、情報収集が終了した際に実行されるプログラムを記述する。

```
<exec> program </exec>
```

と記述すると集計データを標準入力として program が実行される。program はまず集計データを解析し、何からの処理をする。

以下では現在作成できる質問形式について述べる。情報収集サーバが質問票を作成することで、さまざまな質問形式に対応できる。現在の WWW ページを利用した質問票の大部分は、単一回答形式、複数回答形式、自由回答形式の質問が占める。ゆえにこれらの質問形式を用意すれば、現在の多くの情報収集に対応できる。rics はこれらの質問に加え、アナログ回答形式を提供している。

単一回答形式は選択肢の中から一つ選択する質問形式である。質問票定義コードでは図 A.1 のように記述し、図 A.2 のように表示される。

```
<simple_answer>
<question> Simple Answer Question </question>
<choice> choice1 </choice>
<choice> choice2 </choice>
<choice> choice3 </choice>
</simple_answer>
```

図 A.1: 単一回答形式の質問票定義コード

複数回答形式は選択肢の中から複数選択する質問形式である。質問票定義コードでは図 A.3 のように記述し、図 A.4 のように表示される。

自由回答形式は回答者に自由に回答してもらう質問形式である。質問票定義コードでは図 A.5 のように記述し、図 A.6 のように表示される。

アナログ回答形式は連続量で回答できる質問形式である。質問票定義コードでは図 A.7 のように記述し、図 A.8 のように表示される。

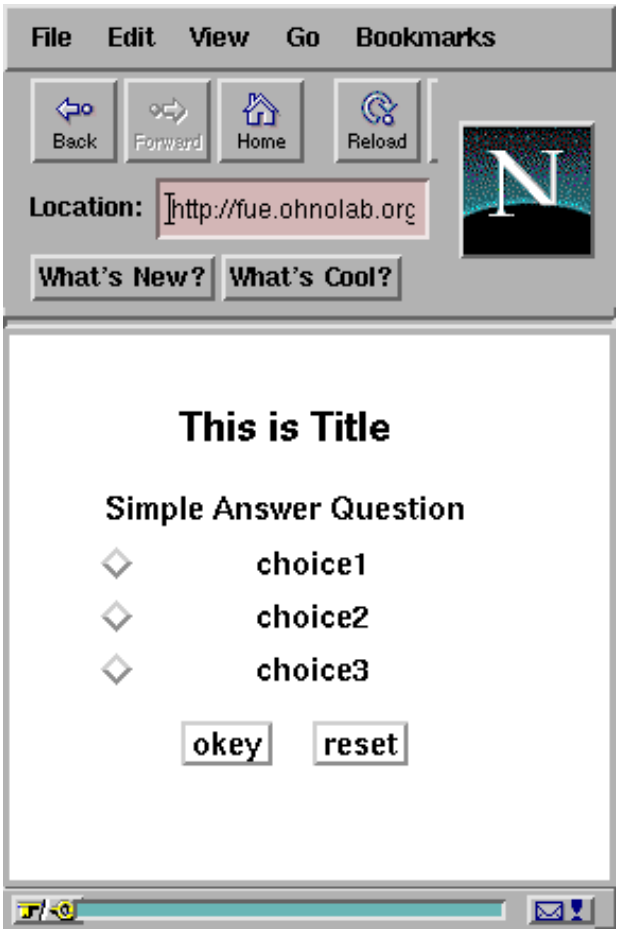

図 A.2: 単一回答形式の質問票

```
<multiple_answer>
<question> Multiple Answer Question </question>
<choice> choice1 </choice>
<choice> choice2 </choice>
<choice> choice3 </choice>
</multiple_answer>
```

図 A.3: 複数回答形式の質問票定義コード

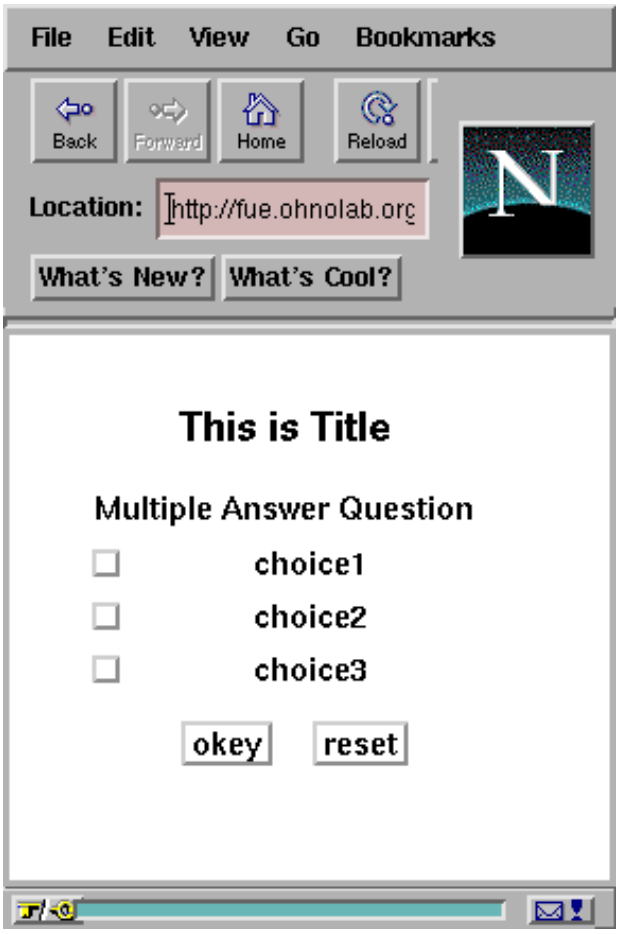

図 A.4: 複数回答形式の質問票

```
<free_answer>
<question> Free Answer Question </question>
<blank width="20" height="1"></blank>
</free_answer>
```

図 A.5: 自由回答形式の質問票定義コード

図 A.6: 自由回答形式の質問票

```
<analog_answer>
<question> Analog Answer Question </question>
<left> Bad </left>
<right> Good </right>
<length> 10 </length>
</analog_answer>
```

図 A.7: アナログ回答形式の質問票定義コード

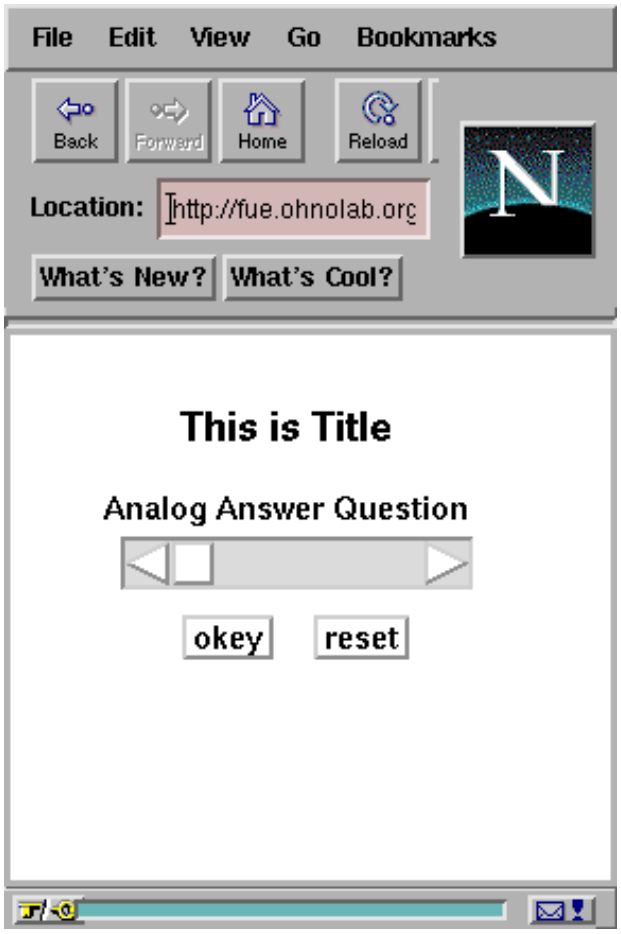

図 A.8: アナログ回答形式の質問票

付 録B　集計サーバと情報収集サーバの動作仕様

この章では集計サーバと情報収集サーバの動作の仕様について述べる。

B.1　集計サーバのコマンド 文法

集計サーバはクライアントからの要求をコマンドとして受け付ける。以下にクライアントからサーバへ送信できるコマンドを示す。

PUBLIC_KEY 公開鍵を送信する。

IAM 引数として QUESTIONNAIRE と ANSWER が指定できる。QUESTIONNAIRE は情報収集クライアントからの接続を示し、rics-send1 プロトコルによる通信である。ANSWER の場合は情報収集サーバからの接続を示し、rics-collect2 プロトコルによる通信である。

集計サーバは IAM コマンドを受け付けると、QUESTIONNAIRE か ANSWER によって異なるコマンドを受け付ける。以下は QUESTIONNAIRE の場合に、クライアントから集計サーバに送信できるコマンドを示す。

FROM 情報収集者の名前を送信する。

METHOD 情報収集形態を送信する。

SIZE 情報収集対象者とする人数を送信する。

GROUP 情報収集対象グループを送信する。

HOST 情報収集サーバのホスト名を送信する。

DATA このコマンドを送信するには、先に FROM,METHOD コマンドを送信していなくてはならない。また情報収集形態 A の場合には GROUP コマンドも必要である。このコマンドを送信後、集計サーバから質問票定義コードを送信するようにとの応答がある。この応答後、クライアントは質問票定義コードを送信する。”.” で質問票定義コードの送信終了を示す。このコマンドを受けた後、集計サーバはクライアントからの情報収集要求を処理する。

QUIT 接続終了。

IAM ANSWER のコマンド場合に、クライアントから集計サーバに送信できるコマンドを以下に 示す。

ID 情報収集 ID を送信する。情報収集 ID については後述する。

DATA このコマンドを送信するには、先に ID コマンドを送信していなくてはならない。このコマンドを送信後、集計サーバから質問票定義コードを送信するようにとの応答がある。この応答後、クライアントは回答データを送信する。”.”で回答データの送信終了を示す。このコマンドを受理した後、集計サーバは回答データを集計する。

QUIT 接続終了。

集計サーバからクライアントへ送信される項目は以下の通りである。

OK コマンドを正常に受理したことを伝える。

NG コマンドを正常に受理できなかったことを伝える。

NO_COMMAND クライアントから未定義のコマンドを受理したことを伝える。

SYNTAX_ERROR コマンドの文法が間違っていることを伝える。

FLAG_ERROR クライアントからサーバへのコマンドには順番があるものがある。コマンドの順番が間違っていることを伝える。

DATA_INPUT DATA コマンドを受理した後、実際のデータを送信するように伝える。

DATA サーバからクライアントになんらかの情報を送信するのに使用される。

COMPLETED クライアントからの要求が正常に処理されたことを伝える。

NOT_COMPLETED クライアントからの要求が処理できなかったことを伝える。

PUBLIC_KEY 公開鍵を送信する。

B.2 集計サーバの状態図

集計サーバの状態図を図 B.1 に示す。

次に、情報収集クライアントからの情報収集要求があった場合の状態図を図 B.2 示す。

情報収集サーバに情報収集の依頼をするときの状態図を図 B.3 に示す。集計サーバは情報収集 ID を作成する。情報収集 ID はランダムかつ唯一であり、回答データを回収するときに使用される。

情報収集サーバから回答データが到着した場合の状態図を図 B.4 示す。

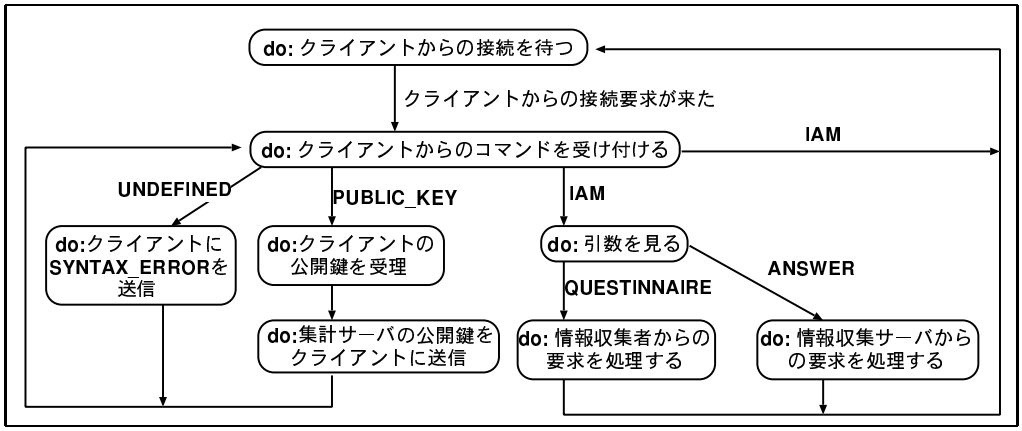

図 B.1: 集計サーバの状態図

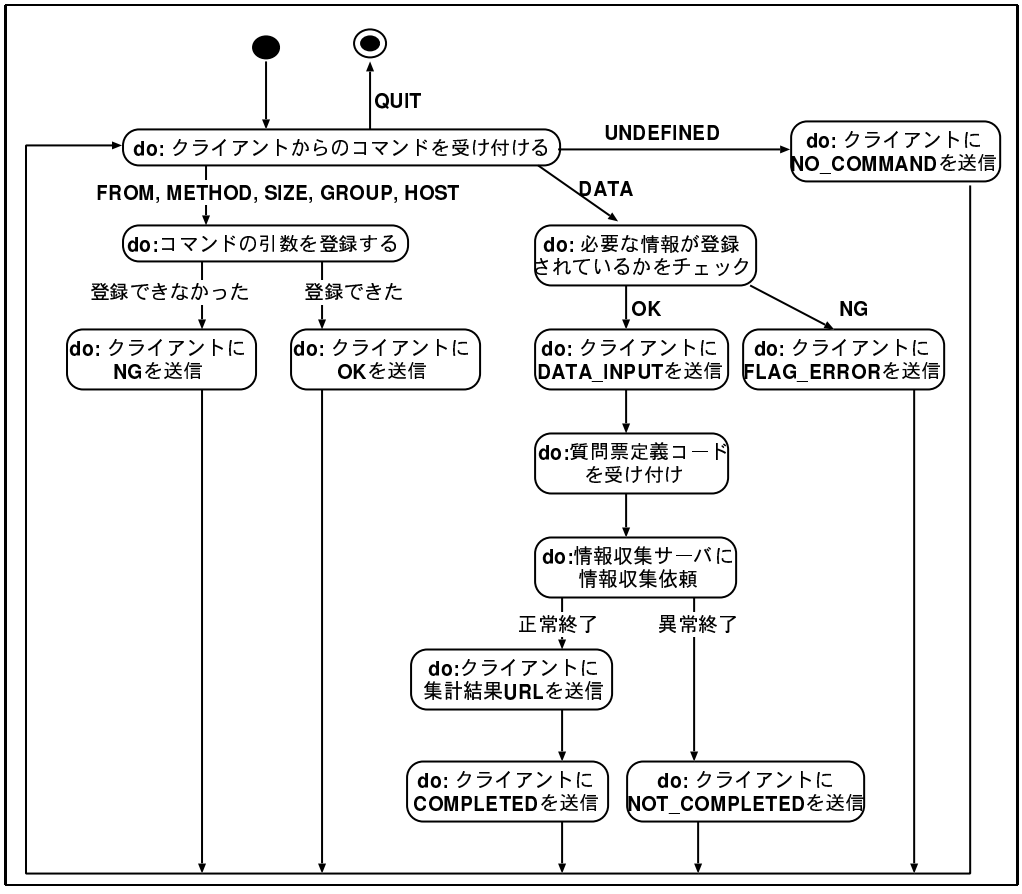

図 B.2: 集計サーバの rics-send1 プロトコルでの通信の状態図

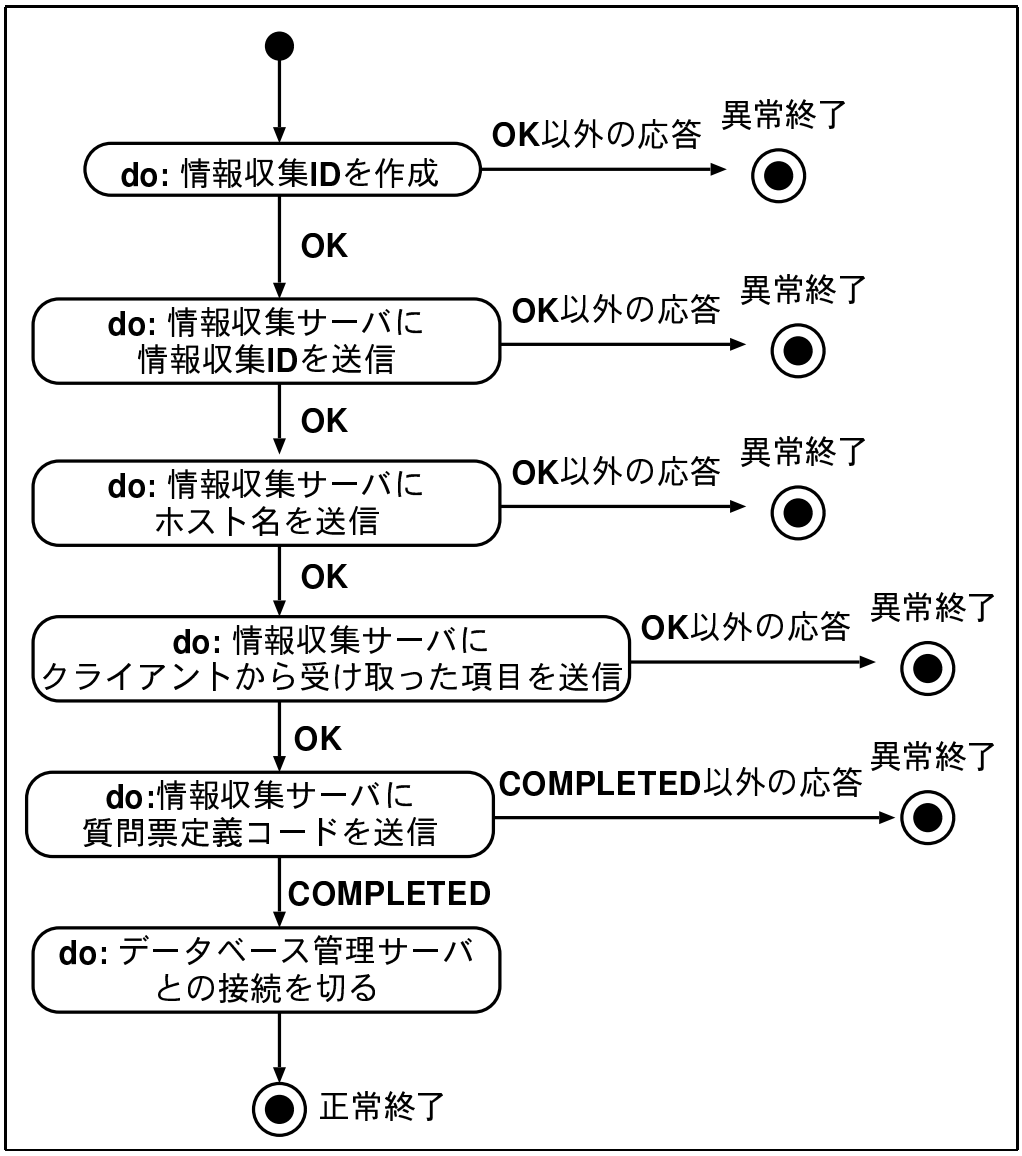

図 B.3: 集計サーバの rics-send2 プロトコルでの通信の状態図

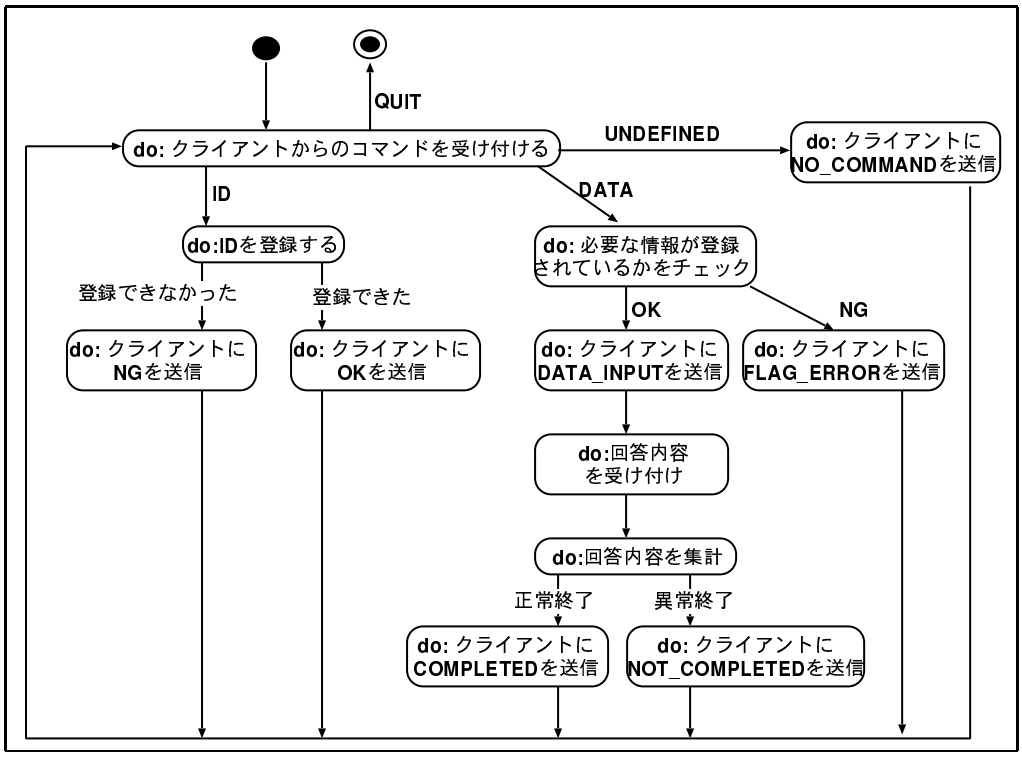

図 B.4: 集計サーバの rics-collect2 プロトコルでの通信の状態図

B.3 情報収集サーバのコマンド文法

情報収集サーバはクライアントからの要求をコマンドとして受け付ける。以下にクライアントからサーバへ送信できるコマンドを示す。

PUBLIC_KEY 公開鍵を送信する。

IAM 引数として QUESTIONNAIRE と ANSWER が指定できる。QUESTIONNAIRE の場合は集計サーバからの接続であることを示し、rics-send2 プロトコルによる通信がなされる。ANSWER の場合は回答処理プログラムからの接続であることを示し、rics-collect1 プロトコルによる通信がなされる。

情報収集サーバは IAM コマンドを受け付けると、QUESTIONNAIRE か ANSWER によって異なるコマンドを受け付ける。QUESTIONNAIRE の場合に、クライアントから情報収集サーバに送信できるコマンドを以下に示す。

ID 情報収集 ID を送信する。

FROM 情報収集者の名前を送信する。

METHOD 情報収集形態を送信する。

SIZE 情報収集対象者とする人数を送信する。

GROUP 情報収集対象グループを送信する。

HOST 集計サーバのホスト名を送信する。

DATA このコマンドを送信するには、先に ID, FROM, METHOD, HOST コマンドを送信していなくてはならない。また情報収集形態 A の場合には GROUP コマンドも必要である。このコマンドを送信後、情報収集サーバから質問票定義コードを送信するようにとの応答がある。この応答後、クライアントは質問票定義コードを送信する。”.”で質問票定義コードの送信終了を示す。このコマンドを受けた後、情報収集サーバはクライアントからの情報収集要求を処理する。

QUIT 接続終了。

IAM ANSWER のコマンド場合に、クライアントから情報収集サーバに送信できるコマンドを以下に 示す。

ID 情報収集 ID を送信する。

HOST 集計サーバのホスト名を送信する。

DATA このコマンドを送信するには、先に ID, HOST コマンドを送信していなくてはならない。このコマンドを送信後、情報収集サーバから質問票定義コードを送信するようにとの応答がある。この応答後、クライアントは回答データを送信する。”.”で回答データの送信終了を示す。このコマンドを受理した後、情報収集サーバは回答データを集計サーバに送信する。

QUIT 接続終了。

情報収集サーバからクライアントへ送信される項目は以下の通りである。

OK コマンドを正常に受理したことを伝える。

NG コマンドを正常に受理できなかったことを伝える。

NO_COMMAND クライアントから未定義のコマンドを受理したことを伝える。

SYNTAX_ERROR コマンドの文法が間違っていることを伝える。

FLAG_ERROR クライアントからサーバへのコマンドには順番があるものがある。コマンドの順番が間違っていることを伝える。

DATA_INPUT DATA コマンドを受理した後、実際のデータを送信するように伝える。

DATA サーバからクライアントになんらかの情報を送信するのに使用される。

COMPLETED クライアントからの要求が正常に処理されたことを伝える。

NOT_COMPLETED クライアントからの要求が処理できなかったことを伝える。

PUBLIC_KEY 公開鍵を送信する。

B.4 情報収集サーバの状態図

以下に情報収集サーバの状態図を図 B.5 に示す。

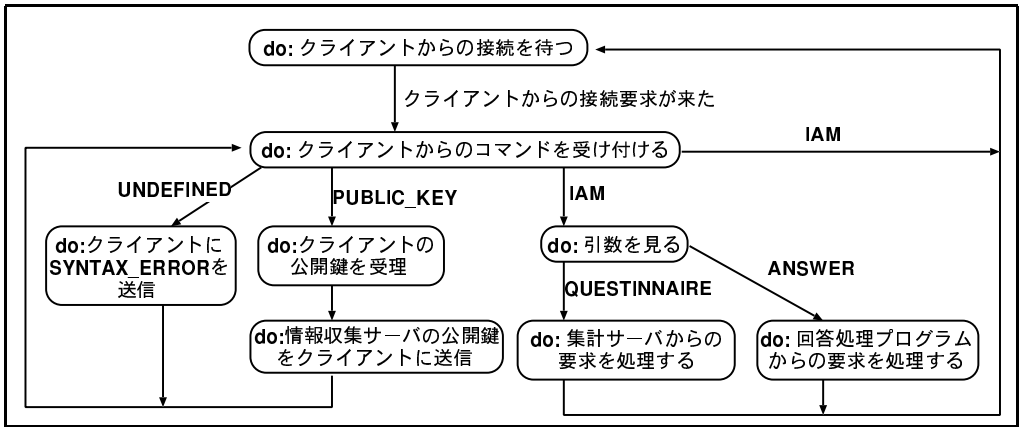

図 B.5: 情報収集サーバの状態図

次に、集計サーバからの情報収集要求があった場合の状態図を図 B.6 示す。

情報収集サーバがクライアントからの情報収集要求を処理する際の状態図を図 B.7 に示す。

情報収集サーバから回答データが到着した場合の状態図を図 B.8 示す。

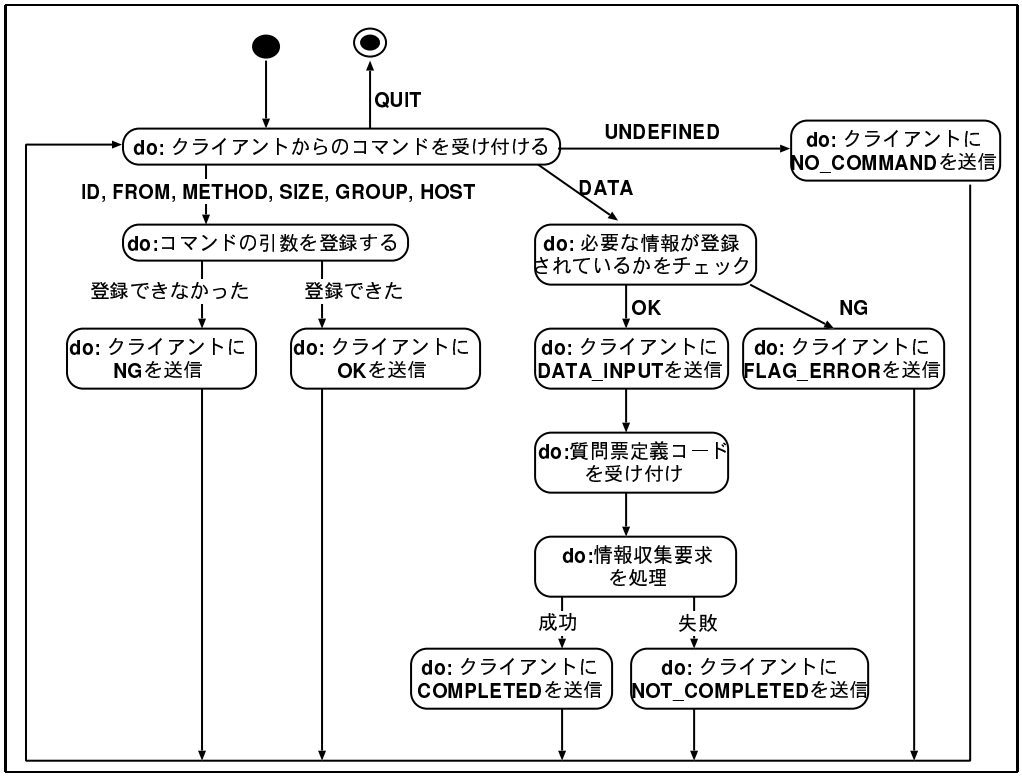

図 B.6: 情報収集サーバの rics-send2 プロトコルでの通信の状態図

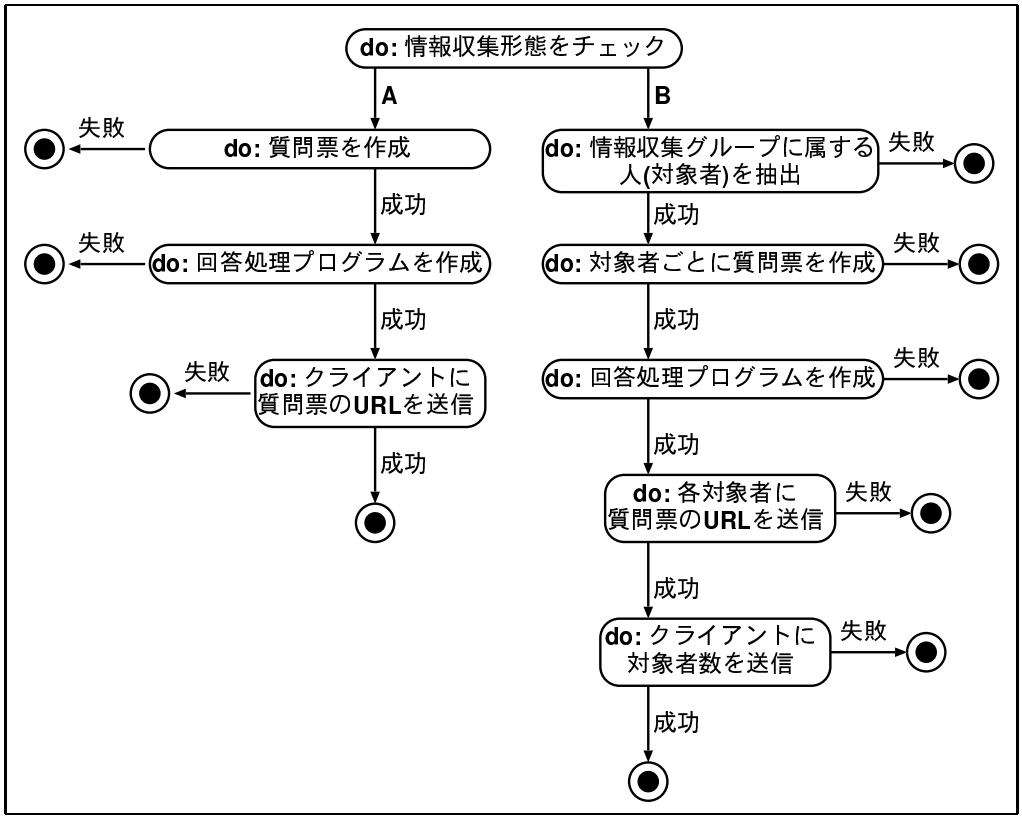

図 B.7: 情報収集の際の状態図

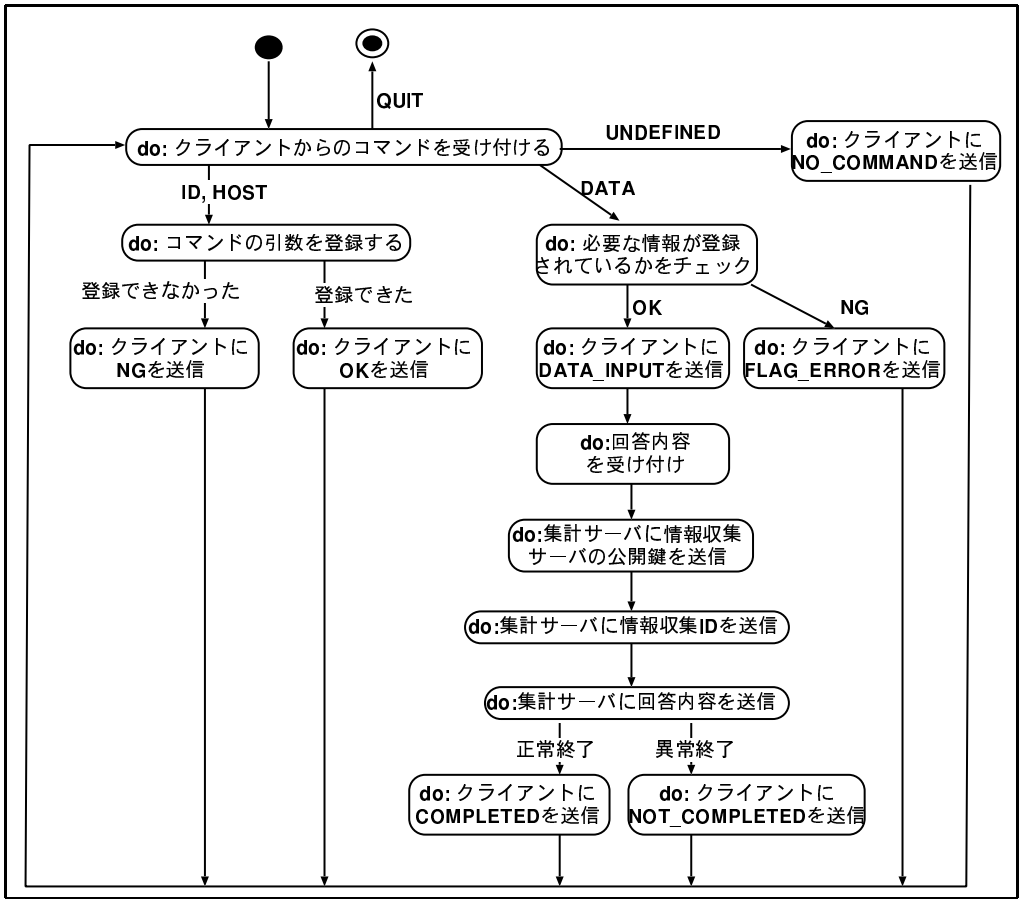

図 B.8: 情報収集サーバの rics-collect1 プロトコルでの通信の状態図