ネットワーク上で協同執筆作業を行なうための
ハイパーテキストシステム

東京工業大学
理学部 情報科学科
門間 信行
(9241774)

1995年度卒業論文

指導教官 大野 浩之 講師

1996年2月6日

目 次

1 はじめに 3

2 協同執筆とその問題点 5
- 2.1 本文中で扱う協同執筆の定義 5
- 2.2 弱い結びつきを持ったグループ 5
- 2.3 現実社会における協同執筆の例 6
- 2.4 協同執筆の三つの形態 8
- 2.5 オンライン協同執筆の効果 9
- 2.6 WWW を用いた協同執筆 10
- 2.7 まとめ 11

3 ハイパーテキストと協同執筆 12
- 3.1 ハイパーテキストとは 12
- 3.2 協同執筆に対する効果 12
- 3.3 協同執筆を考慮したハイパーテキストシステム 13
  - 3.3.1 データの保存方法 15
  - 3.3.2 アクセス制御とデータ更新 15
  - 3.3.3 バージョン管理 15

4 もんじゅシステムの設計 17
- 4.1 概要 17
- 4.2 作業モデル 17
- 4.3 ハイパーテキストシステムの設計 18
  - 4.3.1 Dexter モデル 18
  - 4.3.2 データ保存層 20
  - 4.3.3 ストレージ層の設計 20
  - 4.3.4 ユーザ対話層の設計 23
- 4.4 協同作業機能 24
  - 4.4.1 アクセス制御とデータ更新機能 24
  - 4.4.2 バージョン管理 24
  - 4.4.3 状態変化に伴う著者間の連絡 24

5 もんじゅシステムの実装方法 25
5.1 概要 25
5.2 今回の実装について 26
5.3 差分作成部 26
5.4 解析部 29
5.5 実行部 30
5.5.1 データベース管理プログラムの実装 31
5.5.2 協同作業機能の実現 32
5.6 まとめ 33

6 考察・課題 34
6.1 考察 34
6.2 今後の課題 35

第1章

はじめに

日々の知的生産活動の必需品となったコンピュータを個人的な作業のみならず、複数の人による協調活動に活用する動きが注目されている。日常的に行われる協調活動の例として、多人数が協力して文書を作成する協同文書作成作業がある。従来、協同文書の作成は小規模のグループによって行われ、完成した文書はグループの総意に基づくものであるという感覚が強かった。したがって計算機支援の研究の対象としても、このような小規模のグループの存在を前提としていた。

一方、近年のコンピュータネットワークの整備にともない、ネットワークは多くの人々に利用されるようになりつつあり、新しいコミュニケーションの場として期待されている。ネットワーク上の人々を対象としたさまざまなアプリケーションによって、人々が弱い結びつきながらも自然にグループを形成するような場面がしばしば見受けられるようになった。つまり、ネットワークでは弱い結びつきを持った人々による協調作業が行われる環境が形成されつつある。

しかしながら、ネットワーク上には多くの人々が存在するために、さまざまな情報を持った人による協力が期待できる反面、悪意を持った人による妨害が危惧される。また、基本的に人々の結び付きが弱いために、生成される協同文書がグループの総意であるという感覚は薄い。従来の協調作業支援システムでは、広くネットワーク上の人々を対象としてはいなかったため、これらの点については考慮されていない。

そこで本研究では、執筆作業の一層の分散化を図るために、協同文書作成の対象を、ネットワーク上で弱い結び付きを持った人々に拡大する。人々に広く参加を呼びかけ、弱い結びつきによるグループの形成を促進するための手段として、ネットワーク上の蓄積型の情報発信システム、特に World Wide Web の存在を念頭に置き、遠隔分散したユーザ間による協同執筆を支援する環境の整備を目標とする。

まず、現実社会で行われている協同執筆形態や、すでにネットワーク上で行われている簡単な協同執筆形態を観察し、結び付きの弱い人々による協同執筆について、満たすべき条件について考察する。その上で、ハイパーテキストの参照モデルである Dexter モデルに基づき、ネットワーク上の人々によって書かれた協同執筆文書を細かい単位で保存し、共有空間を実現するための新たなハイパーテキストシステムの設計を行った。

システムに盛り込む機能としては、特に以下の2点に注目した。

- ネットワーク上の人々を対象としたアクセス制御による、文書の細かい単位の管理
- 変更作業による文書の動的な更新と、それに伴う協同作業者への状態変化の通知

前者に関しては、従来の World Wide Web では境界があいまいであった、文書の実体と、リンクやアンカーなどのハイパーテキスト情報とを分離させることによって、文書の細かい単位のアクセス権を明確にした。また、作業を通知する非同期通信のための手段として電子メールを用い、電子署名の技術によって得られる認証機能や改ざん防止機能をシステム内に取り込むことで、アクセス制御の問題の解決を図った。

後者については、作業者が行った文書の変更に基づく、文書の更新作業を自動化することによって、文書の動的な変更を可能にした。さらに、文書の更新によって影響を受ける協同作業者に対して、状態変化を通知する必要性がある。これは更新の際に起こるハイパーテキスト情報の変化の仕組みを工夫し、リンクやアンカーを柔軟に扱うことによって達成を試みた。

最後に、World Wide Web とのインターフェースを定義し、既存のシステムとの連係を図った。

以上により、遠隔分散したユーザによる、World Wide Web 上での協同執筆を可能にし、執筆作業の分散化を実現した。

この論文の構成は以下のとおりである。次章では、協同執筆の定義を行い、現実社会で行われている協同執筆の形態について何点か例を挙げて説明する。その上で、ここで扱う協同執筆形態を限定する。第3章では、ハイパーテキストシステムの設計にあたり、協同執筆の面からどのような点に留意すれば良いのかを述べる。第4章では、もんじゅシステムの設計を行う。第5章では、もんじゅシステムの実装について述べる。最後に、第6章では、まとめおよび今後の課題について述べる。

第2章

協同執筆とその問題点

この章では、協同執筆の形態について述べ、協同執筆作業のモデル化を行う。また、協同執筆の例をいくつか挙げ、本論文において対象とする範囲を限定する。

2.1 本文中で扱う協同執筆の定義

「協同執筆」とは、複数の人が一つの文書の作成に携わるときに用いられる言葉である。協調作業の一環として日常的に行われる協同執筆の一形態として、原案の作成、メンバーによるコメントづけ、修正、といった行程を何度も繰り返すものが挙げられる。

従来、計算機支援の研究の対象となっていた協同執筆とは、このような比較的小さな規模のグループの存在を前提とし、最終的にグループの総意に基づく文書を作成する、といった意味合いが強かった。

本研究では、協同文書を「一つの文書内に、複数の人の書いた部分が混在する場合」とし、協同文書の作成を協同執筆と定義する。さらに、次節以降で述べるように、WWWなどの情報発信システムを意識して、作成された文書は広く公開されることを前提とする。また、文書の作成に関わる対象は特定のグループに限定せず、後述する弱い結びつき持ったグループとする。ここでは特に、対象となる人々はコンピュータネットワーク上に存在しているものとする。

2.2 弱い結びつきを持ったグループ

近年のコンピュータネットワークの整備にともない、ネットワークは多くの人々に利用されるようになりつつあり、新しいコミュニケーションの場として期待されている。ネットワーク上の人々を対象としたさまざまなアプリケーションによって、人々が関心のある情報に集まるようになった。このような事例として、

- あるメーリングリストに所属する人々
- ネットニュースの同じニュースグループを読む人々

- WWW の同じページを閲覧する人々

などが挙げられる。上記した人々は、通常お互い知り合いではなく、他人の存在に気づかないことさえあるが、何らかの共通の関心によって結びつけられている。本研究の対象は、このような「弱い結びつき」を持ったグループである。特に、今回は WWW の存在を意識する。

このような、弱い結びつきを持ったグループによる協同文書の身近な例としては、

- 同じ新聞を講読する人々によって作られる投書欄
- 駅を利用する人々による伝言板
- 郵便局を利用する人々による掲示板

などが挙げられる。それぞれについて、どのような仕組みで作業が行われるを次節で述べる。

2.3 現実社会における協同執筆の例

ここでは、前節で挙げた、現実社会における弱い結びつきをもった人々による協同文書について、それぞれどのように協同執筆が行われ、協同文書が形成されるかについて述べる。

駅の伝言板

駅などの待ち合わせ場所によく設置されている、黒板のようなものである。特定の相手に対してのメッセージという意味合いが強く、誰でも書き込み自由であり、誰でも読めるという特徴がある。

書き込みを行う人は、特に何の制限もなしに、自由に書き込むことができる。自分が何者であるかは、伝言する相手にだけわかればよく、特に身分を明らかにする必要もない。

いたずら書きなど、伝言板の主旨にそぐわない内容が書き込まれた場合には、駅員が判断して書き込みを消す。また、書き込まれてから、あらかじめ決められた時間を経過すると、駅員が消す。

作業の流れは次のようになる。

1. 執筆
2. 執筆者本人による掲載
3. 管理者による事後承認
4. 管理者による抹消

郵便局やスーパーなどで見かける掲示板

郵便局やスーパーなど、人の集まる場所で見かけるもので、地域の住民を対象にしている。「家庭教師引き受けます」「ネコあげます」といった内容が掲示される。

掲示を希望する人は、自分で掲示したい内容を紙に書き、係員へ提出する。身分を明らかにすることが多いようだが、それほど厳密ではない。

係員は、その場で内容を見て、掲示板の主旨にそぐわないと判断した場合には、掲示希望者にその旨を告げ、書き直しを促す。掲示して差し支えないと判断すれば、掲示を許可し、承認の判を押して掲示希望者に返す。

掲示希望者は、返却された紙を、掲示板の好きな位置に貼る。あらかじめ決められている掲示期間が過ぎると、係員が剥がす。

作業の流れは次のようになる。

1. 執筆
2. 執筆者による管理者への掲載の申請
3. 管理者による承認
4. 執筆者による掲載
5. 管理者による抹消

新聞の投書欄

新聞や雑誌など、広く大衆の目に触れるメディアにおいて、読者の意見を伝えるために用いられる。

読者は投書内容を書き、新聞社へ郵送する。新聞は社会的責任が重いため、投書は郵送に限定し、匿名による投書を受けつけないことが多い。新聞社では、一度に掲載できる投書の数には限りがあるため、寄せられた投書の内容を審査し、何通かを選び出す。さらに写植を行い、発行となる。前述の伝言板や掲示板と異なり、新聞は紙に印刷して複製を配布する出版物であるため、抹消を行うことは不可能である。

作業の流れは次のようになる。

1. 執筆
2. 執筆者による管理者への掲載の申請
3. 管理者による承認
4. 管理者による掲載
5. 管理者による抹消

以上のように、多くの人の目に触れ、伴う社会的責任が重いほど、掲載する文書の扱いが慎重になり、管理者による承認の必要性が増す。さらに、新聞のような出版の形式になると、文書中のどの部分に執筆者が担当した部分が掲載されるか、つまりはレイアウト構成を、執筆者自身が決定する権限がなく、管理者によって決定される。まとめると、表2.1のようになる。

表 2.1 現実社会における協同執筆

	グループの規模	承認の必要性	掲載を行う人
駅の伝言板	小	無	執筆者
郵便局の掲示板	小	有	執筆者
新聞の投書欄	大	有	管理者

2.4 協同執筆の三つの形態

Preece らによると、文書の協同執筆には、三つの形態があるとされる [Pre94]。この三つとは、執筆作業の分担による場合、レビューの繰り返しによる場合、及びこの二つが混合して行われる場合である。

これらは、グループ内で行われる協同執筆についての記述であるが、この論文で扱うネットワーク上の人々を広く対象とする協同執筆においても、基本的には同様の形態が考えられる。

以下、それぞれの場合について述べる。

執筆作業の分担による場合

まず一つめの形態は、あらかじめ文書をいくつかに分けておき、それらの執筆協同作業者に分担させる方法である。この形態では、執筆者と編集者が存在する。

執筆者は、自分の担当する部分を書き上げ、原稿を編集者に送る。編集者は集まった原稿を一つの文書にまとめあげる。すなわち、次のような手順を踏む。

1. 協同作業者による打ち合わせ、分担作業の決定
2. 協同作業者各自による執筆
3. 郵便やＦＡＸなどによる編集者への送付
4. 文書の取りまとめ
5. 完成

文書のレビューの繰り返しによる場合

二番目は、執筆者としての権限を一人に持たせる形態である。この形態では、すなわち、一人が作成した原案に対して、協同作業者が加筆および校正を行う。そして、それを基に、一人が原案を書き直す。すなわち、作業は以下のような流れとなる。

1. 原案の作成
2. 郵便やＦＡＸなどによる協同作業者への送付
3. 協同作業者によるコメントづけ

4. 郵便やＦＡＸなどによる原案作者への送付
5. 最初へ戻る
6. 完成

上記の二つの形態が混合して行われる場合

三番目の形態として、上記した二つの形態が混合して行われる場合である。例えば、一番目に述べた執筆作業を分担した場合では、できあがった文書に対してレビューが行われる。また、二番目に述べたレビューによる場合では、文書の部分を分担して校正を行う、といった具合である。

実際には一つの文書を作成するときでも、この二つの形態は混合して行われることが多いとされる。したがって、協同執筆作業において、この二つの形態は連続的に繰り返し現れるものであるため、どちらか一方をサポートするだけでは不充分であり、両方をサポートする必要がある。

2.5　オンライン協同執筆の効果

ここでは、オフラインでの協同文書作成の際に起こる問題についていくつか挙げ、オンラインでの協同執筆によって解決が期待される点について述べる。

まず、オフライン、特に紙文化での協同執筆では、作成した文書の改ざんに気をつけなくてはならないことが挙げられる。新聞への投書を例にとると、投稿された文書を新聞社が適当に誤字・脱字を訂正したり、要点だけを切り出したりしてから掲載することがある。そのため、実際に掲載される内容が、元々の文書と微妙に異なってしまい、投稿者が本来意図していた内容が十分に伝わらなかった、といった事例がある。これは、投稿内容が書かれた紙が、そのまま新聞に載らないために起こる問題である。つまり、紙に書かれた文書の扱いが不連続であるため、悪意がなくても内容を改ざんされたりする事態が起こり、また作業効率も悪くなるのである。

投稿された内容に不備があった場合は、新聞社で勝手に訂正せずに、投稿者に差し戻して不備を指摘し、訂正・再投稿の後に掲載する、という方式を採るのが理想的であると考えられる。

しかし、現実には、投稿者との連絡の必要性にともなう、郵送や書類作成にかかる手間や時間の制約から、その方式を採ることができないことが多い。そのため、新聞社側が適当に文書を修正してから掲載する方式を採るのであろう。

一方、上記した問題はオンラインネットワーク上で行われた場合、解決が期待される。例えば電子新聞を発行しているとすると、投稿された文書をそのままの形で新聞に張り付けることができるので、紙文化で起こっていた不連続性が解消される。改ざんに対しては、電子署名などの技術を応用することによって防ぐことができる。

また、投稿内容に不備があった場合も、電子メールなどの通信手段によって紙文化よりも早期に問題解決を図ることが期待できる。

2.6 WWWを用いた協同執筆

WWWのような広域情報発信システム上で、協同執筆環境を構築することを考える。

WWW上でよく見かける簡単な協同執筆の例として、「訪問者リスト」がある。これは、読者が「訪問者リスト」の登録用のページにアクセスし、自分の名前、メールアドレス、ホームページのＵＲＬ、各種コメントなどを決められた形式に従って入力すると、入力した情報が「訪問者リスト」として書き込まれるものである。

「訪問者リスト」は、純粋にネットワーク上の人々のコミュニケーションを活発化させる作用が期待できるほか、読者側は自分のページの宣伝などに利用でき、ページの開設者側としては自分のページを訪れた人がいかに多いかを宣伝できる、といった特徴がある。

「訪問者リスト」では、書き込みに関して特に何の制限も設けず、全く誰でも自由に書き込みができるようになっているところが多い。このように読者に自由に書き込みを許す場合には、書き込み内容がすぐに掲載され、その場で確認できるために、読者側から見れば気軽に安心して書き込みができる。反面、ページ開設者側から見れば、次のような問題点がある。

- 書き込みを行った人を特定できない。
  WWWでは、通信の相手を特定することは難しい。さらに代理(proxy)サーバを中継して書き込みが行われた場合など、書き込みを行った人を特定するのは非常に難しくなる。

- いたずらの書き込みをされる可能性がある。

一方、上のような問題を回避するために、他人からの意見その他を全て電子メールで受け付けるような方法を採っている事例もよく目にする。例えば、あるページに関する意見をページ開設者に電子メールで送ると、ページ開設者がそれを読み、該当ページにメールの内容を掲載する、といった具合であり、「電子メールを用いた投書掲示板」と呼ぶことができる。

この場合、自由に書き込みを許す場合と比較して、次のような利点がある。

- 書き込み内容が一度ページ開設者を通り、承認されてから実際に掲載されるため、承認のない場合に比べて安全性は向上する。

- 電子メールでは、発信者のメールアドレスや途中の配送の経路などがはっきりわかるため、相手を特定しやすい。

逆に、以下のような点が欠点として挙げられる。

- 書き込みの内容がすぐには掲載されない。

- 投稿が活発化しない可能性が高い。

実際、ネットニュースには、モデレートされた、つまり管理者による承認を経てから記事が配送されるニュースグループがあるが、投稿する人は極めて少ない。

しかし、ネットニュースと違ってWWWのページは基本的に開設した人の持ち物であり、自分の開設したページの主旨に全くそぐわないような書き込みは許したくない。また、書き込み内容が他人の名誉を傷付けているような場合、これは他人の書き込みとはいえ、自分のページに掲載されているということで、ページ開設者に責任が及ぶことも考えられる。

現在では、インターネットに流れる著作物の扱いや、情報に対する責任に対しては明確なガイドラインが示されているとはいえない状態である。しかし、WWWなどのオンライン情報発信システムでは、公開した情報の複製が容易であるため、一度公開した情報を完全に撤回することが困難である。また、公開した情報は不特定多数の人に閲覧される。このことから、WWWでの情報発信は放送や出版のような性格を持っており、社会的責任は軽くない。むやみに争いに巻き込まれるような事態を避けるには、やはりページ開設者は書き込まれる内容を一度確認してから掲載したほうが無難であると思われる。

2.7 まとめ

本研究において対象とする、弱い結びつきのグループでの協同文書作成作業では、グループが集まる場が用意されており、場の管理者によって協同文書が管理されている。多くの人の目にふれ、社会的責任が強い場においては、協同文書の扱いが慎重であり、一度管理者が内容に目を通してから掲載される。

WWWを場として協同執筆を行う場合は、WWWの社会的性格から、掲載内容は一度管理者が内容に目を通してから掲載されることが望ましい。

第3章

ハイパーテキストと協同執筆

3.1 ハイパーテキストとは

ハイパーテキストは、関連する文書の速やかな参照を行うための技術である。

近年は、テキストに加えて静止画や音声、動画などのマルチメディアデータをも扱えることを強調して「ハイパーメディア」と呼ばれることもあるが、ここではハイパーテキストとハイパーメディアは同義語として扱う。

ハイパーテキストシステムは、以下のような特徴を持っている [AMY88]。

- 情報は、「ノード」と呼ばれる小さな単位に分割される。
- ノード間は「リンク」によって結びつけられる。利用者は、ノードからノードへリンクをたどることによって情報の中を航行する。
- 「アンカー」はノードの部分を示す。
- ノードやリンク、アンカーのような、ハイパーテキストシステムにおいて情報間のつながりを表す情報を「ハイパーテキスト情報」と呼ぶ。

3.2 協同執筆に対する効果

協同執筆では、2章で述べたように、協同執筆者は文書の一部分に対する加筆や訂正などの変更操作を行う。

既存の文書に対する加筆や訂正などの操作は、ハイパーテキストにおけるノードやリンク、アンカーの概念を用いるとうまく表現できる。

例えば、ある既存の文章の途中に、自分が何か加筆した場合、図3.1のようになる。

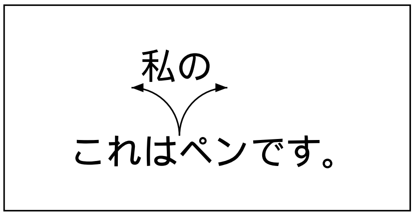

図3.1 文書への加筆

この例では、「これはペンです。」という文に対して、「これは」と「ペン」の間に「私の」を挿入することによって「これは私のペンです。」という文に変更している。

この構造は、図3.2のようなハイパーテキスト情報によって表される。すなわち、自分で書いた文書の先頭からリンクを出し、そのリンクは既存の文書の先頭から、加筆される部分の末尾までを指す。そして、自分の書いた文書の末尾からもリンクを出し、こちらのリンクは既存の文書の残りの部分を指すようにする。自分で書いた部分と前後のリンクを合わせて参照すれば、加筆したように見える。

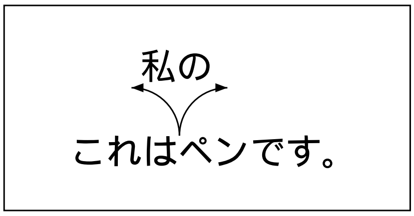

図3.2 加筆を表すハイパーテキスト構造

他にも、文書の一部の削除(図3.3)や上書き(図3.4)、交換(図3.5)といった、文書編集における操作は全てリンクやアンカーなどのハイパーテキスト情報を変化させることで表すことができる。

以上の理由から、複数の人によって書かれた文書を管理するデータベースとして、ハイパーテキストの概念が有効である。本研究での目的となるシステムの開発は、ハイパーテキストの概念を基盤にして行う。

3.3 協同執筆を考慮したハイパーテキストシステム

ハイパーテキストを用いる場合には、いくつかの留意すべき点がある。これらはハイパーテキストシステムの性格を決定づけるものとして重要であるために、以下に述べ、広域ネットワーク上での協同執筆という点から考察する。

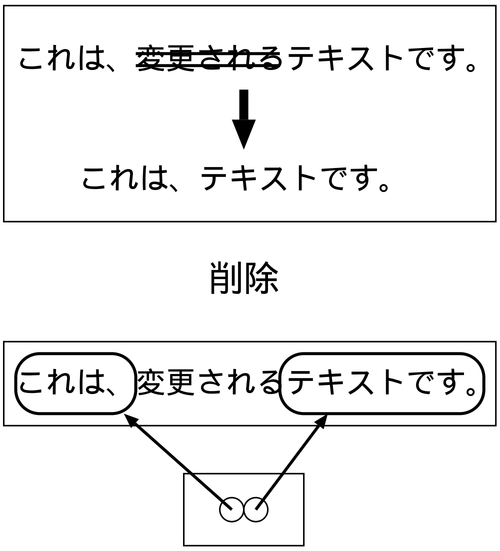

図 3.3 削除

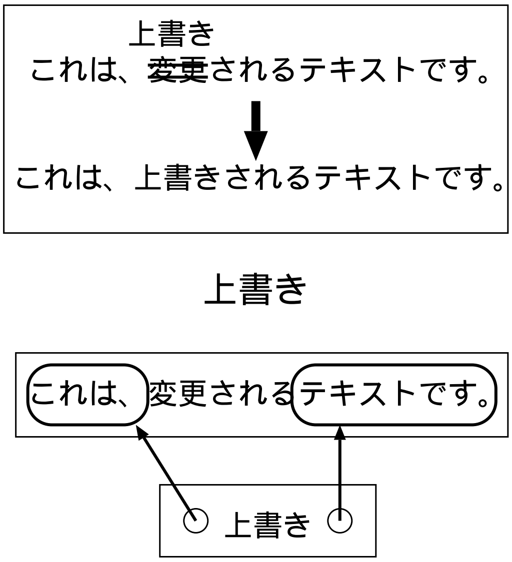

図 3.4 上書き

3.3.1　データの保存方法

協同文書をネットワーク上に展開する方法について述べる。

執筆者が自分の執筆した文書を各自ローカルに保存し、管理者はそのリンクなどのハイパーテキスト情報のみを管理する方法が考えられる。

この方法では、書かれた文書は執筆者の責任で保存されるため、文書の管理としての仕組みは簡単になる。しかし、ネットワークの現況を鑑みると、このように文書の部分をネットワーク上に分散して展開した場合、後に文書の全体を参照する際にいくつものサイトにアクセスする必要があり、コストがかかりすぎる。また、文書の整合性を保つためのバージョン管理機能も、複数のサイトにわたって展開される必要があり、複雑になる。

そこで、ここでは執筆された文書を管理者の手元に集め、ハイパーテキスト情報を含めて全て管理者の元で管理することを中心として考える。この場合は、他人の執筆した文書の管理やアクセス制御に対してより敏感になる必要があり、この点における管理者の負担をいかにして軽減するかが鍵となる。

3.3.2　アクセス制御とデータ更新

現行の WWW では、アクセス制御はファイル単位であり、事実上文書単位となっている。協同執筆のためには、アクセス制御が文書単位ではなく、もっと細かい単位で行われる必要がある。さらに、ここではネットワーク上の人々からのアクセスも考慮しなくてはならない。また、今回扱う協同執筆では、協同文書はグループの総意ではない。従って、執筆者が担当した部分はそれぞれの個人の著作物として扱われるべきである。

3.3.3　バージョン管理

各人の加えた変更の整合性を保つための細かいバージョン管理機構が必要となる。また、ハイパーテキストを広域ネットワーク上に展開するのであれば、複数のサイトのまたがった文書の整合性を保つために、より高度な技術が要求される。

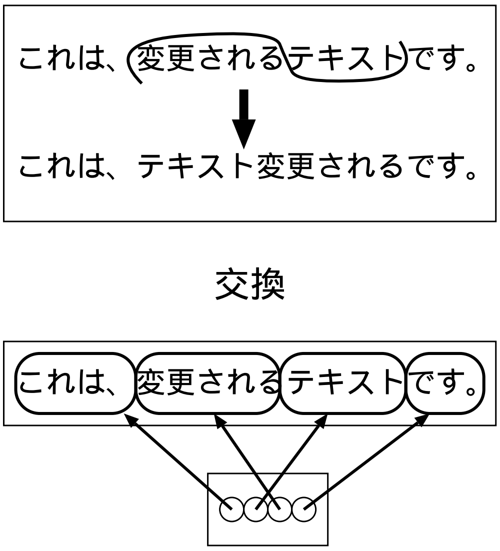

図 3.5 交換

第4章

もんじゅシステムの設計

広域ネットワーク上の人々が協同執筆を行うための仕組みとして、もんじゅ[1] システムの設計を行う。以下に順を追ってこのシステムの内容を説明する。

4.1 概要

このシステムでは、従来単独の人が行っていた文書作成作業を、広域ネットワーク上の人々が参加することによって実現される協同執筆をサポートする。協同作業者は既存の文書を「変更」することで作業に参加する。すなわち、既存の文書の一部に対して加筆や訂正を行う、あるいはあらかじめ示された章立てやレイアウトに従って、空白部分に自分の文書を埋め込む、といった具合である。

協同作業者は、まず公開されている既存の文書を入手し、変更を加える作業を行う。次に、これを元の文書に反映させるために管理者に連絡して承認を得ると、元の文書に反映される。文書の変更に伴うハイパーテキスト情報やデータの動的な変更を行うために、Dexter モデルに基づいた新たなハイパーテキストシステムを構築する。

4.2 作業モデル

ネットワークを介して作業を行う協同作業者は、前節で述べたように、既存の文書に変更を加えることによって作業に参加する。変更作業は以下の手順に従って行われる。

1. 差分の作成
   協同作業者は、ハイパーテキスト上に展開され公開された元の文書を取り出して自分の作業領域にコピーし、それを元にして変更後の文書を作成する。最後に、元の文書との差分として保存する。
2. 変更の連絡

---

[1] これは、日本のことわざ「三人よれば文殊の知恵」からとったものである。別に筆者の名前をもじったものではないし、ましてや高速増殖炉とも関係ない（と筆者は思っている）。ちなみに、名付け親は佐々木毅氏である。

1. で作成した変更前後の文書の差分を元の文書の所有者に送付し、変更の許可を得る。もんじゅシステムでは、連絡手段として電子メールを用いる。このとき、改ざん防止のため、文書の差分の各構成要素ごと電子署名をほどこす。

ここで、連絡手段として電子メールを採用した理由を述べておく。電子メールを用いる利点を以下に挙げる。

- ネットワーク環境にいる人の大多数が利用可能と思われること。また、多数の人にとって最も日常的に利用するネットワーク上のコミュニケーション手段であり、操作に慣れ親しんでいると考えられること。
- 構造化メールや電子署名技術など、電子メールに関する既存の技術を利用しやすいこと。

逆に欠点としては、電子メールでは基本的にテキストデータしか送付できない、大きなデータの送付には向かない、などが挙げられる。しかし、これらの点に関しては、MIME[BF93][Moo93] やその周辺の技術によって解決が図られており、特に大きな障害にはならないと考えられる。

以上の理由から、連絡手段として電子メールを採用した。

3. 変更の承認
   管理者は、2. で受け取った差分を見て、変更を受け入れるかどうかを判断する。承認の必要性については、2 章で述べたとおりである。
4. 文書の差し替え
   3. において変更が承認された場合、システムは差分を元に変更後の文書を復元し、新しい版として差し替える。

以上に述べた手順に従って、協同作業者が変更した文書が管理者の元へ届けられ、元の文書に反映される。

4.3 ハイパーテキストシステムの設計

協同執筆では、最終的に作成された文書内に複数の人がそれぞれ担当した部分が入り混じることになる。この複雑な構造をハイパーテキスト情報で表し、さらに文書中の細かな単位のアクセス制御とバージョン管理機構によって、それぞれの作業者の責任や権利の境界を明確にしたハイパーテキストシステムが必要になる。

そこで、協同執筆を考慮したハイパーテキストシステムとして、ここではハイパーテキストシステムの参照モデルである Dexter モデル [HS94] を基盤としたシステムを設計する。

4.3.1 Dexter モデル

Dexter モデルの特徴の 1 つとして、ハイパーテキスト環境を 3 つの層に分割し、それぞれの層の振る舞いを明確にしたことが挙げられる。このシステムではこれにならう。

文書の実体を保存する部分をデータ保存層、リンクやアンカーなどのハイパーテキスト情報のみを取り扱う部分をストレージ層、さらに、ユーザとのインタラクションを受け持つ部分をユーザ対話層と呼ぶ。各層にはそれぞれの担当する部分に応じて層に対する操作が定義される。まとめると、次のようになる。

1. データ保存層
   Dexter モデルでは Within-Component 層と呼ばれる層である。文書の実体であるテキスト・画像・音声などを保存する部分であり、ファイルシステムと直結している。次に述べるストレージ層からの要求に従って、文書を取り出したり、保存したりする操作が定義される。
2. ストレージ層
   同じく Storage 層と呼ばれる部分であり、文書の構造を表すリンクやアンカーなどのハイパーテキスト情報を管理する。ここでは、新たな文書の導入や文書の抹消などに伴うハイパーテキスト情報の変化を扱う操作が定義される。
3. ユーザ対話層
   同じく Run-Time 層と呼ばれる部分である。ハイパーテキスト情報とユーザとのインタラクションを受け持つ。つまり、ハイパーテキストを画面に表示してユーザの目に見えるようにしたり、ページを開く、リンクをたどる、などのユーザからの要求を受け付けたりする。

この三つの層の概念図を図 4.1 に示す [HS94]。

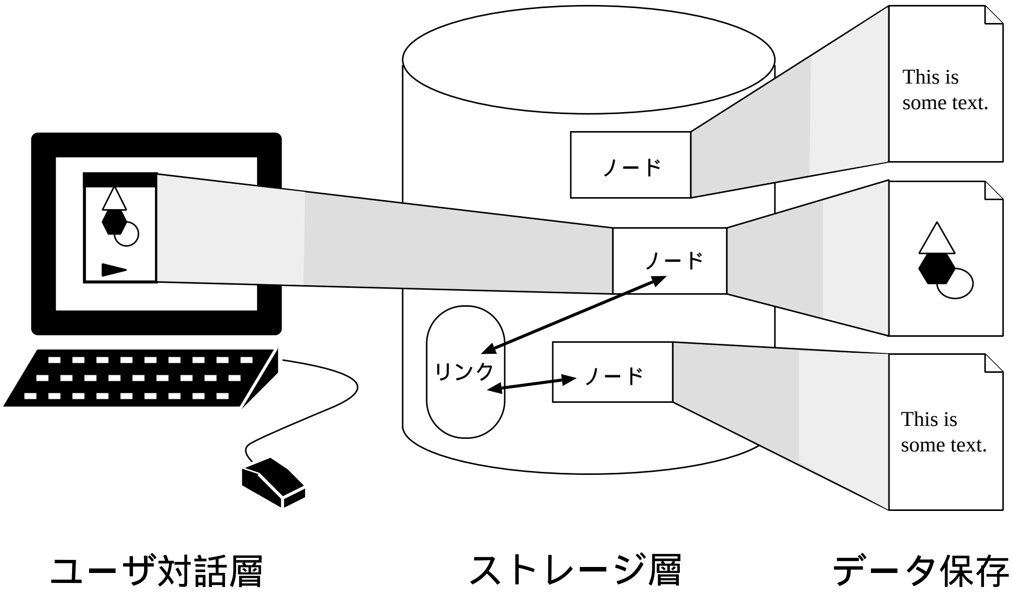

図 4.1 三層のハイパーテキストシステムにおける概念図

このモデルに従うと、協同執筆の面から以下のような利点があると考えられる。

- 文書の実体(データ保存部)と、ハイパーテキスト情報(ストレージ部)が分離しているので、文書の実体を変更することなくハイパーテキスト情報だけを変更することができる。つまり、協同作業者による文書の変更をハイパーテキスト情報の変更だけで行うことができるように工夫すれば、元の文書の実体に触れる必要がない。従って、それぞれの著者の権利や責任の境界を明確にするのに役立つ。

- リンクの表現に柔軟性を持たせることができる。リンクの両端にあるアンカーの付け方を工夫することで、文書の一部分だけを参照することが容易になる。これは、別々の作者によって書かれた文書を自由に組み合わせて参照することに適している。

- 3章で述べたように、文書の変更は、文書内に「ここからここまで」というように「印」をつけ、その範囲で削除・挿入などの操作を行うことに相当する。ここで、印をアンカーとみなして削除・挿入された文書とリンクさせる、といった操作により、文書内に複数の人の書いた部分が混在する状態を表現することができる。

以上の点から、上記のモデルに従い、それに協同執筆のために必要な機能を盛り込む。以下に各層の振る舞いをより詳しく述べる。

4.3.2　データ保存層

テキスト・静止画・音声などの構成要素を単位として文書を分割し、別々に保存する。これらはハイパーテキスト上で別々のノードとみなされる。1つの文書内に複数の著者によって書かれた部分が存在する場合は、それらもまた別々のノードとみなし、別々に保存される。これらのノード群は参照の際には1つの文書として見えるように、ストレージ層においてリンク付けされる。

この層はOS付属のファイルシステムを用いれば十分である。

4.3.3　ストレージ層の設計

基本データ型

Dexterモデルでは、ストレージ層に管理される基本データ型としてリンク、アトム、コンポージットが、また、リンク・コンポージットに対して部分を指定するためのデータ型としてアンカーが定義されている。ここでは、そのままこれにならう。各データ型は、それぞれ属性値を持つことができる。

四種類のデータ型について、順に説明する。

アトム

アトムは1つのテキスト・静止画などの単位である。アトムの大きさは全く自由で、テキストを例にとると、長編小説全体であることもあれば、ただ1文字であることもある。

ただし、実体はデータ保存層に保存されるので、アトムはそれを指し示すことができればよい。

アトムには後述するアンカーを付けることができる。アトムは、要求によって取り出されたときに、後述のリンクの所在を示すために自分につけられたアンカーの情報を取り出す必要がある。従って、アトムは自分に付加されたアンカーのリストを持つ。

コンポージット

コンポージットは1つ以上のアトムを組み合わせたもので、文書全体や一つの章などのまとまった単位を構成し、WWWでのページに相当する。ユーザからのアクセスは基本的にこのコンポージットを単位として行われる。

このシステムでは、コンポージットはアトムを一次元的に並べたものとして表される。

リンク

次に述べるアンカー同士をつなぎ合わせ、アトムやコンポージットの間を結びつける役割を果たす。アトムやコンポージットにつけられたアンカーからリンクをたどると、別のアンカーに導かれ、異なるアトムやコンポージット間を移動することができる。

アンカー

アトムに対して部分を指定する。表現方法としては、

- テキストに対するアンカー

  テキストの先頭から数えてx文字目から始まるy文字の領域

- 長方形の静止画に対するアンカー

  座標(x1,y1)と座標(x2,y2)を頂点とする矩形領域

- 音声に対するアンカー

  再生開始x秒後からy秒間の、ある音域

などのようになる。

アンカーは前に述べたリンクの両端につけられるもので、アトムやコンポージットとリンクの間の橋渡し役を担う。1つのアトムには複数のアンカーをつけることができる。

新しくリンクを張るには、必ずリンクの両側にアンカーを付けなければならない。逆に、アンカーはリンクを張るときのみ必要となる。従って、アンカーは必ずリンクに対応する。

アンカーの役割としては、リンクを導く入口となる場合、すなわち出リンクを持つ場合と、リンクから導かれて出口となる場合、すなわち入リンクを持つ場合の2通りある。

以上4つのデータ型には全て、協同執筆に必要なアクセス制御のために、属性として「著者」を持たせる。それら1つ1つを作成する操作を行った人が著者である。他に共通の属

性としては、テータの取りだしにのために一意的に定められる ID、及び文書間の整合性を保つために必要な情報として、それぞれのデータの作成時刻が挙げられる。

データに対する操作

この層で行われる操作は、以下のようになる。新しい文章の導入などの操作は、これらの操作の組み合わせで実現される。基本的に、削除や変更ができるのはそのデータの著者のみとなる。

- アトム・コンポージットの取り出し

  アトム又はコンポージットを1つ指定して、その指し示すデータをデータ保存層から取り出し、ユーザ対話層へ渡す。データ中にリンクがある場合には、その所在を明示する。取り出す対象としてコンポージットが指定された場合は、コンポージット内にあるアトムを先頭から順に全て取り出すことになる。

- アトムの作成・削除

  アトムの作成・削除を行う。作成の場合は、アトムの実体となるデータと、作成するアトムを指し示すための ID を受け取り、データの実体をデータ保存層に格納する。削除する場合は、アトムにつけられたアンカー、そのアンカーに対して張られたリンクも存在の意味を失うために、消去することになる。このとき、後述するように、消去の対象となるアトムと、それに付属するアンカー、リンクとの著者が異なる場合、アンカー、リンクの著者に対して、リンク先のアトムが抹消された旨の連絡を行う。

- コンポージットの作成・削除

  コンポージットの作成・削除を行う。作成の場合は、既存のアトムを特定する ID の列、作成するコンポージットを指し示すための ID を受け取る。削除の場合は、アトムの削除の場合と同様、当該のコンポージットの消去によって意味を失うデータを削除する。

- リンクの作成・削除

  リンクの作成・削除を行う。この操作には、必ず後述のアンカーの操作を伴う。なぜなら、リンクを張るには、必ずリンクの両側にアンカーを付けなければならず、アンカーに対しての操作は必ずリンクへの操作に対応するからである。作成の場合は、リンクの両端のアンカーを指定する。削除の場合は、消去されるリンクの両端にあるアンカーも同時に消去される。

- アンカーの付加、消去

  アンカーの付加・消去を行う。この操作には、必ず前述のリンクの操作を伴う。付加の場合には、アンカーが付加されるアトムを特定する ID と、アトムの部分を指定するための情報が渡される。消去の場合には、消去されるアンカーにつながれていたリ

ンク、及びそのリンクがつないでいたもう一方の端のアンカーも同時に消去の対象となる。

- 属性値の取得・変更

  各データ型に付加された属性値の取得や変更を行う。

以上の操作により、ストレージ層において管理されるハイパーテキスト情報を変化させる。

4.3.4 ユーザ対話層の設計

この層では、ユーザへの情報の提示とユーザからの要求の受付を担当し、ハイパーテキストブラウザとして実装される。このシステムでは、他に文書の変更を行うための各種エディタが必要となる。ユーザへの情報の提示では、テキストや静止画などの表示、及び表示データ中のアンカーとリンクの所在の表示を行う。実際の画面でユーザの目にどのように見えるかは、ハイパーテキストブラウザの独自性に委ねられる。

ユーザからの要求としては、新たなページを開く、リンクをたどるといった通常の操作の他に、このシステムでは文書の変更作業が重要な役割を持つ。

4.4 協同作業機能

ここでは、もんじゅシステムにおける協同作業の機能について述べる。

4.4.1 アクセス制御とデータ更新機能

このシステムでは、協同作業者が変更を施す際、管理者の元で公開されている文書を適宜自分のローカルサイトにコピーし、個人的な作業空間としてそこで全ての変更作業を行う。従って、変更途中の作業内容は、一切他人には見えない。

変更が終わったら、変更前後の文書の差分を取り、管理者の元へ送付する。これは、文書を再び共有空間に差し戻すことに相当する。

管理者は、差分を受け取り、その変更を受け入れるかどうかを判断する。ここで、変更を受け入れなかった場合には、差分が変更者に差し戻されるので、変更側は自分の作成した文書を独自に公開しても構わない。

4.4.2 バージョン管理

差分を元に差し替えが起こった場合、文書は実際に上書きされるのではなく、ハイパーテキスト構造を変化させることによって上書きされたように見せることになる。従って、元の版はそのまま残され、参照することも可能である。

4.4.3 状態変化に伴う著者間の連絡

ハイパーテキスト上での協同執筆方法の1つとして、他人の文書を自分の文書内に引用する形式がある。このとき、引用する側は相手に対して引用の許可を取り付けるために連絡をとるだろう。これは、このシステムでは相手の文書にリンクやアンカーの変更を加えて差分を取り、相手に連絡することに相当する。

さて、後になって引用される側が参照されている文書を何らかの理由で抹消した場合、引用する側の文書は価値を失う。そこで、再び双方が連絡を取る必要が生じるのだが、このシステムでは参照先の文書を抹消する場合、自動的に参照元の著者へ通知することができる。

また、文書に変更を繰り返すうちに、古い版の存在価値がなくなった場合などは、管理者の判断により古い版を抹消することになるだろう。この場合も、古い版の作成に係わった著者に抹消が通知される。必要ならば、抹消される著作物をその著者に引き取ってもらう、などの処置をとることができるだろう。

第5章

もんじゅシステムの実装方法

5.1 概要

もんじゅシステムは、以下のようなプログラム群から構成される。

- 差分作成部

  変更者が、変更前後の文書の差分を生成し、管理者に電子メールを送る作業を支援する。

- 解析部

  差し換え作業を支援する。送付された差分を元に、データベース管理プログラムに命令を与える。

- 実行部

  ストレージ層のデータを管理するデータベース管理プログラムである。データに対する操作を提供する。常駐し、サーバとして機能する。

これを、前章で述べた作業過程にあてはめると、以下のようになる。

1. 差分の作成
   協同作業者は、実行部に命令を与えて変更すべき文書を取り出し、自分の作業領域にコピーする。これを元に適当なエディタを用いて変更後の文書を作成し、差分作成部のプログラムに通すことで、変更前後の文書の差分を得る。
2. 変更の連絡
   1. の作業の際、差分作成部はそのまま電子メールで送付可能な形式で差分を作る。こうして得られた差分を元の文書の所有者に送り、変更を連絡する。
3. 変更の承認
   管理者は 2. で送付された差分を受け取り、変更を受け入れるかどうかを判断する。
4. 文書の差し替え

3. で変更が承認されると、送付された差分は解析部に渡される。解析部では、差分の内容を解析し、変更後の文書へと差し替えるために実行部に与える命令の列を作成する。
実行部では、これらの命令を受け取り、それに従ってハイパーテキスト情報を変化させ、結果として文書に変更が加えられたように見せる。

5.2 今回の実装について

差分作成部、解析部、実行部の3つのプログラム群のうち、今回は実行部のみについて試作を行った。実行部の実装は、NCSA httpd 1.3 を拡張し、不足するプロトコルやデータ型を追加する方法を採った。
以下に、今回の実装について述べる。

- 他のサーバとの連携は考慮しない

  広域ハイパーテキストの研究では、手元にある文書だけでなく、世界中の文書に対して等しく取り扱うことができることが理想とされ、複数のサーバが連携し合うことで、その実現がはかられている。

  しかし、今回の実装では、自分の手元にある文書について、自由な変更を許可する点の実現を重視した。よって、自分が保存・管理している文書についてのみ面倒を見ればいいのであり、他のサーバとの連携については今のところ考慮しない。

- テキストデータのみを扱う

  システムで扱うデータ型を、テキストデータのみに限定する。

  後に考察でも触れるが、さまざまな形式のデータ型を一括して扱うのは困難である。特にアンカーの指定の方法が難しい。これはデータベース管理プログラム内のデータ構造、さらにアンカーづけのためにデータベース管理プログラムが外部に提供する操作の仕様にも影響を与える。

  そこで、今回は以下のような措置を採った。

  - 今回の試作では、テキストデータのみを扱うものとする。
  - テキスト以外のデータ形式の取り扱いに関しては、今後の課題とする。
  - ただし、将来のテキスト以外のデータ形式への対応を考慮に入れ、テキストデータ固有の挙動を示す部分に関しては、そのことを明記し、拡張の余地を残す。

5.3 差分作成部

差分作成部のプログラムは、引数として変更前後の2つの文書を示すファイル名と、それらを参照するための名前をとり、それらの差分を作成する。

前述のとおり、テキストデータのみを扱うものとする。

2つのテキストデータの相違を得るための代表的な方法として、UNIXのdiffコマンドを用いることが挙げられる。diffコマンドは、行単位で2つのテキストデータを比較し、新しく挿入された行、削除された行、内容が書き換えられている行、の3種類の変更を検知し、結果を出力するコマンドである。

しかしながら、diffコマンドではテキストを行単位で区切って比較を行うため、得られる差分も行単位となることが問題となる。これは、もんじゅシステムでは、別々の人が書いた部分はデータベース内で別々に保存されるためで、差分の作成の際には純粋に元の文書に対して変更された部分だけを取り出す、すなわち行単位でなく、文字単位での処理が必要になる。

従って、差分を得るためにdiffコマンドのような行単位しか扱えないようなツールを用いることは不適当である。

そこで、差分作成部の実装では、テキストデータの比較を、行単位でなく、文字単位で行うようにする必要がある。

動作例

ここで、差分作成部の動作の例を示す。ここでは、後の説明の簡略化のため、元の文書の一部を引用して別の文書を作る、すなわち変更後の文書を元の文書と異なる識別子で参照する例を示す。この例では、識別子としてURL[BLMM94]を用いている。

以下のテキストデータが`http://www/sample`という識別子で示される文書であるとする。

This is a text.

この文書を元にして以下のように変更し、`http://www/sample-mod`という識別子で参照できるようにしたいとする。

This is some text.

この2つの文書を差分作成部のプログラムにかけて生成される差分の例を表5.1に示す。

表5.1 差分作成部が生成する差分の例

```
<BEGIN MONJU SOURCE
URL="http://www/sample">
<a name="anchor1">This is </a>a<a name="anchor2"> text.</a>
<END MONJU SOURCE>
<BEGIN MONJU DIST
URL="http://www/sample-mod">
<a name="anchor3" href="http://www/sample#anchor1"></a>
some<a name="anchor4" href="http://www/sample#anchor2"></a>
<END MONJU DIST>
```

大きく分けて2つの部分から成っている。`<BEGIN MONJU SOURCE URL="...">`と`<END MONJU SOURCE>`に挟まれた部分が変更前の文書、`<BEGIN MONJU DIST URL="...">`と`<END MONJU DIST>`に挟まれた部分が変更後の文書を表す。それぞれの部分はHTML[BLC95]の文法に従っている。

変更前後の文書に共通して現れる部分については、変更前の文書の該当部分にアンカーをつけ、変更後の該当部分からリンクを張っている。

`<a name="anchor1">`の`"anchor1"`はアンカーの名前を示している。ここでは、説明を簡単にするためにこのような簡便な文字列を用いたが、実際にはすでに付加されているアンカー名との重複を防がなくてはならない。そこで、日付や変更者のメールアドレスなどを組み合わせてアンカー名とすることになるだろう。

送付される電子メールの例

このようにして得られた差分を電子メールで管理者へ送付する。送付される電子メールの例を表 5.2 に示す。

表 5.2 変更の連絡のために送付される電子メールの例

```
From:  MONMA Nobuyuki <monma@is.titech.ac.jp>
To:  www-admin@is.titech.ac.jp
Subject:  [MONJU] http://www/sample の変更
X-URL: http://www/sample

  こんにちは。
  http://www/sample を変更しました。
  承認をお願いします。

<BEGIN MONJU SOURCE
URL="http://www/sample">
<a name="anchor1">This is </a>a<a name="anchor2"> text.</a>
<END MONJU SOURCE>
<BEGIN MONJU DIST
URL="http://www/sample-mod">
<a name="anchor3" href="http://www/sample#anchor1"></a>
some<a name="anchor4" href="http://www/sample#anchor2"></a>
<END MONJU DIST>
```

5.4　解析部

解析部では、前出の差分作成部によって生成され、送付された図 5.2 のような差分を入力とする。簡単な構文解析を行い、その結果を内部に保持しているルールと照合して、実行部に与える命令の列を生成する。

図 5.1 で表される差分を解析部のプログラムにかけて得られる命令の列の例を図 5.3 に示す。この例で使われている実行部への命令の書式は簡略化されており、必ずしも実行部の今回の実装における命令の書式と一致しないことに注意されたい。

表 5.3 解析部が生成する命令の列の例

```
ANCHOR sample#anchor1 "This is "
ANCHOR sample#anchor2 " text"
PUT sample-mod "some"
ANCHOR sample-mod#anchor3 0-0
MLINK sample-mod#anchor3 sample#anchor1
ANCHOR sample-mod#anchor4 5-5
MILNK sample-mod#anchor4 sample#anchor2
```

各命令の細かい説明は後に譲るが、例えば 1 行目の

```
ANCHOR sample\#anchor1 "This is "
```

は、`sample` という識別子で得られる文書内の `"This is "` の部分に `anchor1` という名前のアンカーを付加せよという意味である。

また、4,6 行目でも ANCHOR 命令が使われているが、第 2 引数が x-y という形式の数字で示されている。これは、x 文字目から y 文字目までにアンカーを付加せよ、という意味である。この例の場合、4 行目の `0-0` は文書の先頭、6 行目の `5-5` は文書の末尾を表している。この指定の仕方はテキストデータ固有の性質を用いたものであるため、他のデータ型に対する拡張を考慮しなければならないが、この点については後述する。

入力となる差分には、元の文書のどの位置にアンカーを付加すればよいか、新しい文書からどのようにリンクを張ればよいか、などが HTML の文法に従ってすでに記述されている。そこで解析部では、これらの情報を実行部で用意されている命令に置き換えればよい。

5.5 実行部

実行部は、データベース管理プログラムである。外部から命令が与えられた場合、保持している文書に対するハイパーテキスト情報を変化させることで命令に応える。

外部に提供する操作

今回の実装では、すでに広まっている WWW の資源を利用できるようにするため、前述のとおり NCSA httpd 1.3 を拡張する方法をとった。従って、外部に提供する操作も、基本的に WWW での標準的プロトコルである HTTP[BLFF95] に従っている。ただし、HTTP にない命令もいくつか追加している。

外部からの文書の取り出しの命令に対して、文書を外部へ転送する場合、文書間のリンクなどのハイパーテキスト情報はHTMLを用いて記述することとした。ただし、HTMLを用いて表す構造は、アンカーの所在とそのリンク先を示す場合のみである。

情報を取得するためのメソッド

HTTPでは、情報を取得する機能として、GETメソッド及びHEADメソッドさえあれば良いとされている。

GETメソッド

URIで識別されたコンポージット又はアトムを取得する。これは、コンポージット又はアトムに対して取り出しの操作を行うことに相当する。リンクの所在はHTMLのタグを使って表す。

HEADメソッド

HEADメソッドは、サーバにURIで識別されたオブジェクトの属性のみを取り出して返すことを要求する。これは、コンポージット又はアトムに対して、属性の取りだしを行うことに相当する。

変更作業に必要なメソッド

HTTPで提供されるメソッドのうち、PUT、DELETE、LINK、UNLINKの各メソッドについては、それぞれアトムやコンポージットの作成、削除、リンクの作成、削除に相当する。しかし、アンカーに対する操作はHTTPでは提供されていない。このため、アンカーに対する操作に関しては独自にHTTPを拡張する必要がある。

また、HTTPでは、既存の文書への加筆や注釈を扱うためのメソッドとして、POSTメソッドが定義されている。しかし、今回は、変更作業は全て電子メールを基盤とした非同期的通信によって行われるため、POSTメソッドは必要としない。

5.5.1 データベース管理プログラムの実装

ストレージ層で定義される基本データ型であるアトム、コンポージット、リンクおよびアンカーを構造体とみなし、それぞれのデータに対する取り出しや、削除といった操作を関数呼び出しによって実現する。また、アトムを1つのファイルとみなして保存する方法をとった。

アトムやコンポージットを特定するための属性であるIDには、URI(Universal Resource Identifier)を用いている。URIは、RFC 1630[BL94]で定義された包括的資源識別子であり、情報の位置を特定するために使用されている。

今回は、検索の効率化を図るため、アトムを示す URI には、実際にアトムの実体を格納するファイルシステムのパス名を用いた。コンポージットについては特に基準を設けず、自由な文字列を許した。アトム中のアンカーを示すには、アトムの URI の後ろに '#' をつけ、アンカーの名前を続ける。このアンカーの名前を、アンカーを指し示す ID とみなした。

ところで、ハイパーテキスト情報は、柔軟なファイルロック機能やトランザクション機能などの同期的な処理を意識して、オブジェクト指向データベース上に永続的に記憶させる実装法もある [GHMS94]。今回は、電子メールを仲介した非同期的通信を基盤として用いるため、同期的処理の必要性がないと判断し、オブジェクト指向データベースを用いなかった。しかし、オブジェクト指向データベースには、強力な検索機能や、データを永続的に記憶する機能など、魅力的な部分も多いため、オブジェクト指向データベースを用いた実装は今後の課題とする。

5.5.2 協同作業機能の実現

アクセス制御

管理者の元へ送られる電子メール中に含まれる差分に基づいて、サーバはハイパーテキスト情報の変更のためにストレージ層に対して適用する操作を判断するが、アクセス制御については、届いた電子メールから取得される情報を利用する。

アトムその他のデータの作成が行われるときには、届いた電子メールの From: フィールドから、変更後の文書の著者を決定し、属性として持たせる。また、データの削除が行われるときには、電子メールの From: フィールドと、削除が行われるデータの著者を比べ、同一であれば削除を行う。異なっていれば、メールの送信者には削除の権限がないと判断する。

将来は、From: フィールドではなく、付加された電子署名からメールの発信者を特定し、より厳密な意味のアクセス制御を行う必要があるだろう。

状態変化の著者への通知

４章で述べたように、このシステムでは文書の変更や削除などによって他の執筆者の文書に影響が出る場合、関わりのある執筆者に対して通知を行う。例として、アトムの削除が行われると、そのアトムを含むコンポージットに影響が出る。また、アトムに付加されたアンカーも意味を失う。そのアンカーにつながっているリンク、さらにそのリンクからたどられるアンカー、アトム、コンポージットにも影響が出る。従って、以上の影響のあるデータの著者に対して全て通知を行う。通知の手段としては電子メールを用い、削除されたデータからたどられるデータの著者宛にメールが発信される。

HTML でできないこと

もんじゅシステムでは、設計上は文書の同一の部分から複数箇所へのリンクを出すことができる。しかし、ハイパーテキスト記述言語として HTML を用いると、この機能を表現することができない。これは、HTML の能力の限界である。従って、HTML ブラウザにも表現できず、この実装ではこの機能は無駄になる。

5.6 まとめ

以上により、遠隔分散したユーザによる、World Wide Web 上での協同執筆を可能にし、執筆作業の分散化を実現した。

第6章

考察 ・課題

6.1 考察

今回は、実装が不充分であるため、実際の動作に基づく十分な評価が行えなかった。そこで、システムで用いた作業モデル、および設計段階で明らかになった問題点について述べる。

管理者の承認を支援する必要性

今回は、文書の変更を電子メールを用いて管理者へ送付し、管理者が承認してはじめて変更内容が掲載される形態をとった。しかし、この場合、管理者にたくさんの電子メールが届けられ、承認の手間に追われる可能性がある。また、管理者が不在の際、メールを受け取ることができず、その間の更新作業が滞る事態が考えられる。さらに、承認作業は慎重を期するため、軽く扱うことはできない事情がある。

この問題の解決を図るには、承認作業を支援するソフトウェアの開発を行う必要がある。変更前後での内容がよくわかるように工夫したインターフェースや、外出先で電話やFAXからでも承認作業が行えるように、多様な入出力方式への対応が必要であろう。

多様なメディアに対するアンカーの表現

今回用いたモデルにおいて、テキストや画像に対して部分を指定するアンカーは、協同文書において複数の人が書いた部分が混在する状態を表すために欠かせない。しかし、ネットワーク上には、テキストや画像以外にも多種多様なマルチメディアデータが存在するため、その全てに対してアンカーを定義するのは困難である。

ユーザ認証機構の充実

今回の実装法では、電子署名の効果は、差分を送付する際の電子メールの改ざん防止、および管理者の元での記憶された文書の改ざんの防止にしか現れない。また、改ざんが行

われた場合は、確かに検証を行えば改ざんの事実を明らかにすることはできるのだが、文書がアクセスされるたびに検証を行うことは、処理速度の点、および署名用の鍵配布サービスなどの整備状況の未熟さの点から現実的ではない。

したがって、今回用いた改ざん防止のための電子署名は、現在の環境ではその威力を十分に発揮することができない。電子署名をとりまく環境の一層の整備が求められる。

6.2 今後の課題

今回は、ネットワーク上の弱い結びつきを持つ人々による協同執筆を可能とすることを目標として、ハイパーテキストシステムの設計を行った。ここでは、今後の課題について述べる。

1. 実装の充実
2. システムの評価のための実験
3. より厳密なアクセス制御
4. 検索の効率化

前章までに述べたように、3. はより積極的な電子署名の利用により、また 4. はオブジェクト指向データベースによる実装によって改善されることが期待される。その上で、実装を充実させ、システムの評価を行うことを、当面の課題とする。

謝辞

研究を進めるにあたって、丁寧な御指導を頂いた大野浩之講師に心から感謝致します。

また、研究の進め方や論文の書き方について多数の助言を与えて下さった木村 泉教授、木村・大野研究室の先輩方、そしてTitanet運用センターの飯島昭博技官、応用物理学科北原研究室の山下雄三技官、佐々研究室の越塚登助手に、この場を借りてお礼を申し上げます。

参考文献

[AMY88] Robert M. Akscyn, Donald L. McCracken, and Elise A. Yoder. KMS: A Distributed Hypermedia System For Managing Knowledge In Organizations. *CACM*, Vol. 31, No. 7, pp. 820–852, Jul 1988.

[BF93] N. Borenstein and N. Freed. MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies, RFC1521. September 1993.

[BL94] T. Berners-Lee. Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web, RFC1630. June 1994.

[BLC95] T. Berners-Lee and D. Connolly. Hypertext Markup Language - 2.0, RFC1866. November 1995.

[BLFF95] T. Berners-Lee, R. Fielding, and H. Frystyk. Hypertext Transfer Protocol – HTTP/1.0, 1995.

[BLMM94] T. Berners-Lee, L. Masinter, and M. McCahill. Uniform Resource Locators (URL), RFC1738. December 1994.

[GHMS94] Kaj Gronbaek, Jens A. Hem, Ole L. Madsen, and Lennert Sloth. Cooperative hypermedia systems: A dexter-based arhitecture. *CACM*, Vol. 37, No. 2, pp. 64–74, Feb 1994.

[HS94] F. Halasz and M. Schwartz. The Dexter hypertext reference model. *CACM*, Vol. 37, No. 2, pp. 30–39, Feb 1994.

[Moo93] K. Moore. MIME (Multipurpose Internet Mail Extensions) Part Two: Message Header Extensions for Non-ASCII Text, RFC1522. September 1993.

[Pre94] J. Preece. *Human-Computer Interaction*, chapter 28, pp. 588–590. ADDISON WESLEY, 1994.