

パーソナルネットワークに基づく情報流通支援

大向 一輝

目次

第1章	序論	1
1.1	はじめに	2
1.2	Community Web	3
1.3	Community Web における活動モデル	5
1.4	本研究の課題と戦略	7
1.5	本論文の構成	8
第2章	関連研究	10
2.1	協調的タスクスケジューリングの関連研究	11
2.2	メタデータを利用した情報共有プラットフォームの関連研究	13
第3章	パーソナルネットワークを用いた個人リソースの管理モデル	27
3.1	研究背景	28
3.2	タスクスケジューリング問題	28
3.3	協調的スケジューリング	29
3.4	タスクの依頼関係に基づくアクセスコントロール	31
3.5	人間関係ネットワークを用いたアクセスコントロール	33
3.6	本章のまとめ	39
第4章	協調的タスクスケジューラの実装と評価	43
4.1	携帯電話への実装	44
4.2	実証実験	63
4.3	考察	69
4.4	本章のまとめ	71

第 5 章	Weblog とセマンティック Web による個人の情報流通モデル	74
5.1	研究背景	75
5.2	Weblog とセマンティック Web	75
5.3	Weblog のアーキテクチャ	77
5.4	メタデータ層	78
5.5	マネジメント層	79
5.6	アグリゲーション層	85
5.7	アプリケーション層	85
5.8	本章のまとめ	87
第 6 章	個人のための情報流通プラットフォーム	88
6.1	Semblog プラットフォーム	89
6.2	システム構成	92
6.3	PermaRSS	94
6.4	FOAF	97
6.5	Weblog ツールの拡張	97
6.6	FOAF TrackBack	97
6.7	RNA: RSS 収集・配信サービス	98
6.8	RNA Alliance	103
6.9	パーソナルオントロジーの構築	107
6.10	Semblog プラットフォームを利用したコミュニティ支援	110
6.11	本章のまとめ	119
第 7 章	結論	120
	謝辞	123
	参考文献	125
	研究業績	130
付録 A	協調的タスクスケジューラのデータ構造	135

目次

1.1	Community Web における情報・コミュニケーション活動モデル	7
2.1	Weblog サイトの例	14
2.2	ソーシャルネットワーキングサービス「Orkut.com」	25
3.1	協調タスク表示	33
3.2	カレンダー表示	34
3.3	タスク詳細設定	35
3.4	グループの定義と情報共有範囲の拡張	36
3.5	ユーザごとの人間関係ネットワーク	37
3.6	グループ発見アルゴリズム	40
3.7	アクセスコントロールのアルゴリズム	41
3.8	グループ発見を用いたタスク情報のアクセスコントロール	42
4.1	システム構成	45
4.2	カレンダービュー	47
4.3	カレンダービュー（実機）	48
4.4	新規タスク	51
4.5	新規タスク（実機）	51
4.6	文字入力画面（実機）	52
4.7	協調メンバーの変更	52
4.8	協調タスクビュー	54
4.9	協調タスクビュー（実機）	54
4.10	協調タスクビュー（実機）	55
4.11	ユーザマネージャ	56

4.12	メニュー（実機）	58
4.13	メニュー（実機）	58
4.14	同期後の確認ダイアログ（実機）	59
4.15	同期後の確認ダイアログ（実機）	59
4.16	環境設定（実機）	60
4.17	協調タスクスケジューリングの実例	64
4.18	人間関係ネットワーク	73
5.1	RSS 1.0 の記述例	80
5.2	Movable Type におけるコンテンツ入力画面	82
5.3	Weblog ツールのアーキテクチャ	83
5.4	RSS アグリゲータ「glucose」	86
6.1	System Architecture	93
6.2	PermaRSS の記述例	96
6.3	RNA: Snapshot	100
6.4	Glucose: Snapshot	102
6.5	エゴセントリックネットワークの例	104
6.6	エゴセントリック検索	105
6.7	システム動作画面	106
6.8	Personal Ontology Framework	107
6.9	Bottom-up Ontology	109
6.10	JSAI2004 Weblog	111
6.11	学会参加者の Weblog 一覧	112
6.12	発表一覧（Weblog 版）	113
6.13	授業支援システムのスナップショット	115
6.14	経済産業研究所公式サイト	116
6.15	TRIGGERS!公式サイト	117
6.16	goo RSS リーダー	118

表目次

2.1	日記の分類	18
4.1	フィルタリング手法の評価	66
4.2	フィルタリング手法の性能	67

第 1 章

序論

1.1 はじめに

近年のパーソナルコンピュータの高性能化や低価格化，ブロードバンド環境の普及により，インターネットに接続しているユーザの数は爆発的に増加した [1]．それに伴い，インターネットを利用した商業的活動が成長し，これを利用するユーザがさらに増えるという循環が起こっている．今後もこの流れはやむことはなく，ユーザ数，さらには1ユーザあたりのインターネット利用時間は増加し続けるものと思われる．

このようなインターネットのコモディティ化は既存のマスメディアにも大きな影響を与えている．すでに広告の分野では，4大メディアと呼ばれるテレビ，ラジオ，新聞，雑誌に続くメディアとしてインターネットが挙げられている．その広告費の規模はすでにラジオに匹敵すると言われており，数年後にはインターネットを含めた5大メディアとなるとの予想もある．

テクノロジーから見たインターネットの重要性は，それがプラットフォームとして機能するという部分にあると思われる．すなわち，プラットフォーム上では，アプリケーションの設計によってさまざまな情報の流通形態を実現することが可能になる．電子メールに代表される1対1のテキストによるコミュニケーションから，Webのようなテキストと画像による情報公開，映像のストリーミング配信など，利用可能な手段は増加の一途をたどる．また，これらは配信側と受信側が非同期的に情報のやりとりを行うものであるが，リアルタイム性の高いアプリケーションとしては，チャット，IPテレフォニー，映像のブロードキャスティングといったものが実現されている．他にも，利用主体の匿名性の有無，既存メディアとの連携など，アプリケーションの特性によって流通形態は大きく変わり，現在でも技術的，ビジネス的な試行錯誤が続けられている．

メディアの多様化は，インターネットを利用するユーザのコミュニケーション手段だけではなく，伝えるべき内容の多様化にもつながっている．1対1のメディアでは連絡が主だったものが，複数人が参加可能なものでは議論や意思決定のツールとなり，ブロードキャスティング型のは既存のマスメディアと同等の働きをする．また，ユーザ数の増加はユーザ層の拡大につながり，専門家同士のコミュニケーションだけではなく，家族間でのコミュニケーション，あるいは別の専門を持つ人間同士のコミュニケーションといったバラエティが生じつつある．

こういったメディアの進化は，インターネットの持つインタラクティブ性と，そのインタラクティブ性を利用するためのユーザのリテラシーが高まったことにより新たな展開を迎えている．とくに，ユーザがマスメディアの視聴者であった時代と，自らが情報の発信者として行動するようになった現在とでは，コンピュータ，ネットワークあるいはシステムが支援すべき対象の範囲が全く異なる．

Shneiderman は著書 [2] の中で、われわれの思考を”Old computing”から”New computing”へ移行させるべきであると述べている。”Old computing”とは「コンピュータに何ができるか」ということを中心に考えるものであり、”New computing”はそれによって「ユーザにとって何が可能になるか」が関心になるような思考である。Shneiderman は続けて「今後求められるテクノロジーはユーザ側のニーズに調和するものであり、それらは experience を豊かにするためにユーザの持つ『関係』や『活動 (Activities)』を支援するものでなければならない」と述べている。

これを踏まえて、われわれは研究の対象を情報・コミュニケーション技術 (Information Technologies: IT) もしくは Information and Communication Technologies: ICT) から情報・コミュニケーション活動 (Information and Communication Activities: ICA) へ移行すべきであると考えます。ここでは、情報やコミュニケーションに関する人間の活動への理解や、その活動がどのように支援されるべきかを調査する必要があります。

本研究では、情報の受け手であると同時に発信者であるようなユーザ同士がどのように行動し、新たな情報を生み出すかというプロセスをモデル化する。そして、このモデルをインターネット上のメディアに適用し、どのようにメディアを進化させるべきかという指針を提示する。

1.2 Community Web

1989年に誕生して以来、Webの規模は拡大の一途を続けている。初期のWebは研究者間の情報共有ツールとして用いられていたが、ユーザ数が増加するにつれ、企業の広告メディアとしての利用や、Eコマースといった商業利用といった新たな用途が考案されてきた。情報共有についても、アカデミック分野の情報のみならず、個人の興味などに関する情報が続々と公開され、発達途上の検索エンジンやディレクトリサービスと組み合わせることで多くの情報を容易に得られる環境が構築されてきた。また、UsenetやBBSなどのサービスでは、特定の話題に関する参加者同士の議論がサポートされ、これをインターネット上に公開状態にすることでさらに詳細な情報を得ることが可能になった。

10数年に渡る情報の蓄積の結果、Webは巨大な辞書と呼べるほどの情報量を誇るまでになった。大量の情報の中から目的のものを得るためには検索システムの整備が必須であるが、ポータルサイトや検索エンジンが商業的に成立するようになった昨今では、多くのサービスが提供されるようになっている。現在も、より精度の高い情報検索を目指して、Webページ内に含まれるハイパーリンクの構造解析や自然言語処理技術が研究され、実用化されている。このように、Webは情報の集積所として用いられ、ユーザはWebからいかに必要な情報を発見するかということが研究あるいはビジネスの主題となってきた。

一方で、Web にアクセスするユーザが増加するにつれ、新たな動きが顕在化されてきた。それは、Web をコミュニケーションの基盤として利用するというものである。ここでは、Web は現実生活における友人あるいは知人とのコミュニケーションを補完するものとして、会話や議論、あるいはコラボレーションの場として機能している。

これまでの、辞書としての Web においては、ユーザは発信する情報や検索する情報について客観性を求める傾向があった。ニュースや事実に関する情報は、多くの場合において複数の情報源が同様の内容を配信することから、その情報の記名性、すなわちその情報が誰のものであるかについての保証は強くは求められない。しかしながら、Web をコミュニケーションの基盤であると考えられる場合には、情報をやり取りする主体（個人）のアイデンティティの表出、記名性が必須となる。このように、個人の存在が表出するような場を、本研究では「Community Web」と呼ぶ。

Community Web は、個人の現実世界での活動を支援するために、他者とのコミュニケーションを可能にする場である。Community Web が提供すべき機能は、複数人間での協働・コラボレーションを行うにあたっての調整や交渉機能、あるいは個人同士の興味や意見の表明と議論機能であると考えられる。両者には必ずメッセージのやり取り、すなわちコミュニケーションが必要となるが、本研究におけるコミュニケーションの定義を以下に示す。

Community Web におけるコミュニケーションとは、実世界においてユーザ自身を取り巻く友人関係・知人関係（以下パーソナルネットワークと称す）の中で情報が移動することである。パーソナルネットワークは現実の生活を送る中ですでに構築されているものとする。そして、パーソナルネットワーク上で個人同士が情報を交換することで協働のための調整を行ったり、パーソナルネットワーク上の知人から情報を収集し、新たな情報を発信する際に利用する。

情報の流通に際し、一般のコミュニケーション理論においては、情報の送り手は受け手を事前に想定し、受け手に対してメッセージを送付するというモデルや、あるいは送り手が受信可能な対象すべてに一斉にメッセージを送信するブロードキャストモデルが採用されてきた。その一方で、Web は各個人がボトムアップにデータベースを構築し、情報の受け手は検索エンジン等を利用して必要な情報を収集することで、非同期的かつ匿名的な情報流通を実現している。

これらに対し、Community Web における情報流通モデルでは、各個人は Web 上にその個人を代表するサイト、あるいはエンティティを持つ。このサイトには個人に関する情報ならびに個人が生み出した情報がすべて集約され、公開されている。このサイトは必ずしも実名で運用される必要はないが、自己同一性を保証するためにハンドルネームなどが用いられることが多い。

このような環境において、あるユーザは他のユーザに関する情報を収集する。そして、その

情報がコミュニケーションを行うに値するものであると判断した場合、ユーザはその返信を自身のサイトに掲載する。こういった情報は、Web上に存在するために他の誰からもアクセス可能である。一方、返信をすべきユーザがこの情報を読むかどうかについては何ら保証されていない。

ここでは両者の間でコミュニケーションが成立しているかどうかについて判断することは難しいが、十分な数の個人が参加し、互いに得た情報を公開し続けているような状況では、非同期かつ非直接的ではありながらも情報が流通していると考えることができる。

このモデルの利点は、直接的な個人間のコミュニケーションを要請しないために、情報発信に際する心理的な障壁が低いことや、他の個人にも情報が共有されうるということが挙げられる。

本研究では、このような Community Web における情報流通形態を明確に定義し、さらなる活性化を目指すべく、実世界で運用可能なアプリケーションの提案と検証を行う。

1.3 Community Web における活動モデル

本節では Community Web における個人の活動をモデル化し、詳説する。

情報に関する活動やコミュニケーションに関する活動の種類は多岐に渡るが、Shneiderman は Activities and Relationships Table: ART と呼ばれる理解の容易なモデルを導入することで分類を行っている。

ART の一方の軸は”Activity category”すなわち行動の種類であり、”Collect”、”Relate”、”Collaborate”、”Donate”の4種類に分類されている。他方の軸は”Category of relationship”（関係の種類）であり、関与する人々の規模によって”Self”、”Family and friends”から”Citizens and market”に至るまでの4種類に分類されている。本研究では後者の軸には同意するが、前者の”Activity category”については情報を扱うプロセスと人々の間のコミュニケーションプロセスが混同されているため、より詳細な検討が必要であると考えられる。

情報とコミュニケーションのプロセスを検討するにあたっては、複数人の意見を集約し、最終的に1つの結論を出すための意思決定過程に関する理論が有効に機能すると思われる。Simon は、意思決定過程を、事前の情報収集を意味する”Intelligence Activity”（情報活動）、得られた情報から次に取るべき行動の案を設計する”Design Activity”（設計活動）、そして複数の案から1つを選択し、行動に移す”Choice Activity”（選択活動）の3段階に分類している [3]。

このモデルにおいても、各プロセスに情報の生成や行動の決定といった情報のライフサイクルと、それらを生み出す主体である人々のコミュニケーションサイクルが同一化されているが、

これらは分離して考えることが可能である。

一方、コミュニケーションの立場から情報流通を考えた場合には、口コミによる情報探索モデル [4] や個人の信頼の伝搬モデル [5] など、多数の人間が Web 上に存在していることを前提とした新たなモデルが提案されている。これらは、すでに存在する人間関係のネットワーク上で個々のユーザが他のユーザを評価し、どの情報を誰に送信するか、あるいは受信した情報にどれほどの信頼性があるかを判定する手法である。このモデルでは、情報の送り手、受け手の存在が明示化されており、その上で流通する情報が決定されるという 2 層構造を形成している。

このように、情報流通とコミュニケーションは相互に関係し、表裏一体の構造を形成している。これをふまえて、本研究では情報とコミュニケーション、あるいはパーソナルネットワークの関係を明確にするために 2 層の拡張モデルを提案する。概念図を 1.1 に示す。

第 1 の層は情報の扱いに関する 3 種の要素があり、それぞれ”Collect”,”Create”,”Donate”とする。これはユーザを中心とした視点から見た情報のライフサイクルである。情報はユーザによって収集 (Collect) され、それらの情報に基づいて新しい情報が創造 (Create) される。そして新しい情報は社会に提供 (Donate) される。新たな情報が無から作り出されることは稀であり、多くの場合は既存の情報が下敷きとなる^{*1}。これらのプロセスはループをなしており、Donate された結果は次の情報のライフサイクルの Collect に接続される [6]。

第 2 の層はコミュニケーションの扱いに関する”Relate”,”Collaborate”,”Present”の 3 種の要素である。これはユーザ中心のコミュニケーションプロセスであるといえる。ある人物が他の人々との関係を得て、新しい情報を生み出すために協調する。そして彼ら自身が新たな情報源として社会に対しその存在を表明する。それぞれのプロセスは、第 1 層における各プロセスと 1 対 1 対応しているものと考えられる。

情報流通およびコミュニケーションのプロセスは、上記の 6 種のカテゴリによって表現される。これらの活動を支援するにあたっては、対象となる問題がどのカテゴリにあてはまるのかを分析し、その結果をもとにシステムの設計を行う必要がある。

理想的には全てのカテゴリがコンピュータによって支援されるべきであるが、”Collect”のようにすでに研究の蓄積があるカテゴリの一方で、コミュニケーション層に属する 3 種のカテゴリについては、Web のように大規模なユーザを収容できる環境が生まれて間もないため、研究の数は多くない。さらなる取り組みが必要であると考えられる。

本研究では、人間の情報活動およびコミュニケーション活動の調査や分析を行い、その結果

^{*1} 知識の創造が既存のもの組み合わせのみによってなされるということを主張しているのではない。ここでは既存の知識への深い理解と分析が創造のための 1 つのアプローチであると考えている。

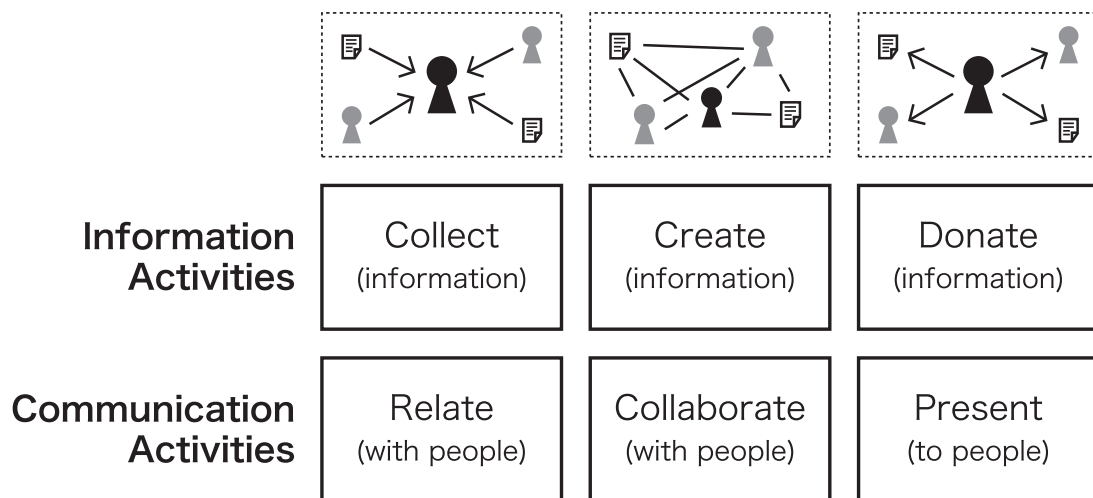


図 1.1 Community Web における情報・コミュニケーション活動モデル

を踏まえた上で全てのカテゴリへの支援を行うことを目指す。

1.4 本研究の課題と戦略

本研究は、これまでに述べてきた Community Web およびその上での活動モデルをコンピュータネットワーク上で実現することを目的とする。実現に際して、以下に示す 2 点が大きな課題として挙げられる。

1 つは、情報のコントロールをいかに緻密に実行するかという問題である。ネットワーク上に個人の存在を表出し、その上で情報を流通させるにあたっては、プライバシー保護の観点から、どの情報を誰に伝えるかを制御する必要がある。また、情報の受け手にとっては、多量の情報の中から必要な情報を選択するにあたり、誰からの情報を受け取るかを定められることが望ましい。

このような情報のコントロールは有益であるが、そのためには事前のプロファイリングを必要とするなど、ユーザ側に多大なコストを課すことが多い。持続可能な情報のコントロールを実現するにあたっては、こういったコストとのトレードオフを考慮する必要がある。

2 つめの問題は、Community Web モデルを実際の Web に適用する際に導入すべきアーキテクチャの形態である。すでに数億のユーザがおり、100 億を超えるページが存在する Web に対して、スケーラビリティを確保できる手法を選択する必要がある。また、ユーザビリティについても、既存の Web のユーザモデルを損ねない形で実現することが望ましい。

これらの問題に対して、本研究では、以下に示す 2 つの個別研究を行った。

情報のコントロールに関する課題については、個人の持つ情報を共有することが問題の解決に大きく役立つ一方で、そういった情報がプライベートなものであるために過剰な情報公開が許されないような対象問題として、個人のタスクスケジューリング問題を取り上げた。この問題では、個人のスケジュールの中に複数人で行うタスクが多く含まれていることから、個人の管理問題であっても複数人で協調的に解決する方がよい結果を得られる可能性がある。しかしながら、個人のスケジュール情報をすべて公開することは現実的ではないために、何らかのコントロールが必要となる。

この問題を図 1.1 のモデルにあてはめると、6 種のプロセスの中でループが形成される部分、すなわち Donate - Collect 間において情報をどのようにフィルタリングするかという問題に帰着する。本研究では、この問題の解法として、Donate および Collect プロセスを支える Present, Relate の両プロセスに注目し、パーソナルネットワークを利用することで、ユーザに負担がかからない情報流通を円滑に実現する方法を考案した。本論文では、この研究を「協調的タスクスケジューリングに関する研究」と呼ぶ。

次に、Community Web モデルを現実の Web の表現活動に適用する際に生じる課題については、まず現在の Web が Community Web 活動モデルの一部をサポートしているに過ぎないことを確認する。そして、これを補うために利用可能な技術として Weblog とセマンティック Web ならびにパーソナルネットワークを取り上げ、これらの融合によってスケーラビリティとユーザビリティの両立を図る。また、実装のために必要なデータ構造の定義や、そのデータを利用するアプリケーションの開発を通じ、提案システムが一般のユーザが実運用可能であることを示す。本論文では、この研究を「メタデータを利用した情報共有プラットフォームに関する研究」と呼ぶ。

研究を実施するにあたっては、実問題を対象とすることを第一義とし、個人が実際に利用可能なソフトウェアあるいはシステムを構築することを重視した。これにはソフトウェアの品質を高めることや、インターフェイスを綿密に設計することなどが含まれる。また、本研究で提案するアルゴリズムやシステムにおいては、実世界で運用可能なスケーラビリティが確保できる手法を選択した。

1.5 本論文の構成

本論文の構成は以下の通りである。本論文は大きく分けて、第 1 の研究である協調的タスクスケジューリングに関する研究と、第 2 の研究であるメタデータを利用した情報共有プラットフォームに関する研究の 2 部構成となっている。第 3 章および第 4 章は前者の研究について述べ、第 5 章および第 6 章は後者の研究について述べる。

第 2 章では，本研究の位置づけを明確にするため，関連研究を示した上で議論を行う．

第 3 章では，個人のリソース管理問題の一種であるタスクスケジューリング問題に焦点を当て，これに関する概説と，協調モデルによる解法の提案について述べる．また，協調モデルにおけるユーザのふるまいを定義し，基本的なアルゴリズムを提案する．第 4 章では，第 3 章で提案した協調モデルを実世界で適用するためのシステムの構築について述べる．また，このシステムの実証実験について述べ，得られた結果より考察を行う．

第 5 章では，情報共有プラットフォームの基盤となる Weblog およびセマンティック Web について概説し，これらを組み合わせたモデルの提案および特徴を述べる．第 6 章では，提案モデルの実現のために，本研究で行ったメタデータ的设计，アプリケーションの実現方法について述べる，また，実装されたシステムを実世界で運用した結果について報告し，これをもとに提案モデルの有効性について検討を行う．

最後に，第 7 章で本論文を結ぶ．

第 2 章

関連研究

本章では，本研究の特色や位置づけを明確にするため，第 1 の研究である Community Web モデルにおける情報のコントロールと協調的タスクスケジューリングに関する研究，および第 2 の研究であるメタデータを利用した情報共有プラットフォームのそれぞれについて，既存のタスクスケジューリングに関する研究，Weblog に関する研究，セマンティック Web，人間関係ネットワーク，知識共有といった研究分野からの関連研究を紹介し，問題点を指摘した上で本研究の特徴について議論する．

2.1 協調的タスクスケジューリングの関連研究

第3章および第4章では協調的タスクスケジューリング問題について取り上げる。この研究に関する第1の関連分野としては工学的スケジューリング問題とその解法が挙げられるが、問題が明確に定義されており、操作的に解決を目指すこれらの研究分野とは目的が異なると思われる。本研究では個人の持つ情報をどのように可視化し、共有するかが大きな課題となっており、その点において CSCW、知識共有システムなどの関連研究が存在する。また、個人を代表するエージェントを利用し、複数のエージェント間の交渉によって自動的なスケジューリングを目指す研究も多い。以下にそれらの概要を述べる。

2.1.1 工学的スケジューリング

一般に、スケジューリング問題において対象となるものは、工場における生産工程の最適化 [7]、交通システムの運行スケジュールの最適化 [8] など、問題の内容が工学的に明確に定義できるものである。こういった問題に対しては、これまでにオペレーションズ・リサーチの分野において線形計画法や動的計画法といった手法の適用が試みられてきた。

問題の定義が厳密でないものに対しては、近年のソフトコンピューティング技術の応用として、遺伝的アルゴリズムを用いたスケジューリング手法 [9] や、ニューラルネットワークとの組み合わせによる最適化手法についての検討が進んでいる [10]。

本研究では、以上の手法が取り扱っているような、あらかじめ問題が定義できるようなものではなく、人々が日常的に行っているスケジューリングを支援することを目指している。このような問題においては、前提となる条件がリアルタイムに変更され、また最適化の基準となる評価関数も個人によって異なる。よって、前述の手法では問題を解消することが困難である。本研究では工学的スケジューリング問題の解法とは異なる手法を提案し、検証を行う。

2.1.2 CSCW とスケジューリング

Computer Supported Collaborative Work (CSCW) は、人々が共同作業を行うにあたってコンピュータや情報システムがどのように支援できるかを検討する学問分野である [11]。CSCW では、コンピュータによる支援手法の提案および人間の共同作業に関する知見の抽出という2つの課題について取り組まれており、グループウェアをはじめとするさまざまなソフトウェアが提案され、検証されている [12]。

CSCW ならびにグループウェアの分野には、会議支援やナレッジマネジメント、意思決定

支援など組織内におけるあらゆる行動の支援が含まれる [13][14] . その中でも組織の構成員のスケジュールリングは大きなトピックであり, 商用のグループウェアである Lotus Notes^{*1} や Microsoft Exchange^{*2} , サイボウズ^{*3} といった製品においても, スケジュール機能は必要不可欠なものになっている. 近年では, 個人単位で容易に導入可能な P2P モデルに基づくグループウェアが開発されており, 事前に設定したグループごとにスケジュール情報を共有することが可能になっている.

これらのソフトウェアは, あらかじめ支援すべきグループの範囲が決定されているが, 本研究が目的としているのはグループの範囲が容易に決定できず, また動的に変更されるような状況における支援モデルの提案である. また, 本研究では複数のグループにまたがる情報共有モデルについても検討する.

2.1.3 エージェントによる個人のスケジュールリング

個人のスケジュールリングを自動的に決定する手法として, マルチエージェントシステムを用いた交渉を行う研究が多数存在する. これらの研究では, グループにおける最適なスケジュールリングと各個人のスケジュールとの調整を主目的にしている点で, 本研究と類似している.

Garrido らの研究 [15] では, ユーザがあらかじめスケジュールに対する好みを設定し, マルチエージェントシステムはこれらの制約を交渉によって緩和することで適切なスケジュールを得る. また, Haynes の研究 [16] では, ユーザは好みと妥結のためのしきい値を入力することでエージェントの挙動を制御する会議用スケジューラを提案している.

Wellman らはスケジュールリングの調整に際し, 各エージェントがリソースを入札することで利害の均衡を目指す市場アプローチの手法を提案している [17].

これらはエージェント間の交渉方式に関する研究であるが, それに対して伊藤らの研究 [18] では, 最終的に得られるグループ内のスケジュールが各個人にとって整合性のあるものかどうか注目し, ユーザの初期設定に確信度を導入することで問題の解決を図っている.

以上はエージェントや最適化アルゴリズムを用いることにより, 各個人のスケジュールが自動的に定まる手法であるが, 一般のユーザからはどのようなプロセスを経て最終的な結果が導かれたのかを知ることが非常に難しい. 本研究では, 状況の可視化やインターフェイスの改善によってユーザ自身が問題の解決に臨めるような環境を提供することを目指している.

*1 <http://www.lotus.com/>

*2 <http://www.microsoft.com/exchange/>

*3 <http://cybozu.co.jp/>

2.1.4 インターフェイス

スケジューリングの支援においては、時間情報を適切に可視化することで問題の発見を容易にすることが可能であると思われる。暦本の研究 [19] では、PC のデスクトップ上で時間情報を直感的に扱うことのできるインターフェイスについて検討を行っている。この研究では直接的にスケジューリング問題を扱っていないが、他の研究成果と融合することでユーザに対して有用な支援が可能になるとと思われる。

インターフェイスの分野では、ユーザの負荷となる情報入力のコストの低減も大きな課題である。乃村らの MHC は電子メールからイベント情報を抽出し、その時間データをカレンダーに自動登録するシステムである [20]。電子メールによるタスク情報の流通は本研究で用いている携帯電話アプリケーションよりも簡便であり、潜在的なユーザ数も大きい。しかし、MHC では複数人の間での情報公開や協調的なタスク登録といった問題を取り扱っていない。

2.2 メタデータを利用した情報共有プラットフォームの関連研究

本論文の第 5 章および第 6 章で述べる情報共有プラットフォームに関する研究では、Web に個人の存在を表出させ、個人間のコミュニケーションプロセスと情報流通を一体化させることを目指す。その際に必要となる技術として、Weblog、セマンティック Web ならびに人間関係ネットワークに関する成果が挙げられる。このうち、Weblog は非常に新しい分野であるために文献が極めて少ない。そこで、本節では Weblog についてとくに詳しく述べ、その後に関各研究の概説を行う。

2.2.1 Weblog の概要

近年の Web におけるさまざまな活動の中で、とくに注目されているのが「Weblog (ウェブログ)」である。1998 年頃から登場した Weblog サイトは、すでに全世界で 1 千万に近い数に達していると言われており、現在も爆発的に増加している。いまや各プロバイダやポータルサイトでは顧客サービスの一環として Weblog のホスティングサービスを提供することが当たり前なものとなった。これに伴う関連サービスも数多い。Weblog は既存のマスメディアやジャーナリズムにも大きな影響を与えており、Weblog 上での議論が世論に反映するような事例も出始めている。このように、Weblog は社会システムとして定着しつつあると思われる。



図 2.1 Weblog サイトの例

その一方で、Weblog サイトは一見してそれと判別できるような特殊な形態をしているわけではない。図 2.1 に示すように、従来と同様の HTML ファイルがハイパーリンクによって接続されたものである。このため、Weblog は新たな名称をつけることによって作為的に起こされた一過性の流行現象でしかないと懐疑的に見る向きも多い。しかしながら、Weblog の普及に際して、コンテンツの記述システムや情報収集の手法など、HTML ファイルを公開するまでのプロセスを支援する技術が飛躍的に進歩している。

Weblog は、Web 上での分散型コミュニケーションを実現する新たなツールとして注目されている。Weblog では、掲示板や Wiki とは異なり、各個人が 1 つずつサイトを持ち、そのサイトの中で自身の意見を記述するものである。そして、Weblog サイトの持ち主同士の議論や意見交換は、相手の持つコンテンツへのリンクや、TrackBack (トラックバック) と呼ばれる逆リンク機構を利用して行うことができる。なお、コンテンツの記述や管理、リンクなどは、Weblog ツールや Weblog サービスを利用することで容易に実現することができる。日本ではすでに 10 万以上の Weblog サイトが存在するとも言われており、それらのサイトで日々身辺雑記や専門知識が公開され、議論が行われている。

Weblog は Web+log からきた造語である。「Blog (ブログ・ブログ)」と呼ばれることも多い。現在では、「Blogging」のような動詞的表現や、「Blogger」といった人を表す言葉も派生している。

Weblog という言葉が定着し、広範なユーザを集めるようになったのは、専用のツールが開発されて簡便にサイトを公開できるようになった 1998 年ごろからである。アメリカで Weblog がまず注目されはじめたのは、新しいジャーナリズムとしての側面であったといわれている。Weblog はマスメディアの制約にとらわれずジャーナリストが自ら意見表明を行うことのできる場であった。その後、9.11 の同時多発テロ後に一般市民が Weblog を通じて意見の交換、議論を行うようになり、そのことが Weblog の認知度を急速に上げた。今日、Weblog はこのような草の根ジャーナリズムのようなコンテンツから、いわゆる日記にいたるまで、様々なコンテンツを提供するための基盤となっている。

Weblog の大まかな歴史については以上の通りであるが、Weblog そのものについての明確な定義は存在しない。一般的には雑記や他サイトへのリンク、それに関するコメントが日々更新されるようなサイトの総称であるとされている [21]。Weblog サイトでは、一定の読者層を想定して体系化されたコンテンツではなく、書き手が興味の赴くままに記述した短いコンテンツを配信する形態となっていることが多い。こういった、頻繁に更新される短いコンテンツのことをここではスモールコンテンツと呼ぶ。スモールコンテンツの内容は多種多様であり、日記から批評、他サイトの紹介などフォーマットも大きく異なる。

比較的多くの人々が受け入れている定義の例として、Blood による著書 [21] での説明を挙げる。Blood は Weblog の形態をブログ、ノート、フィルタに分類したうえで、

- ブログ
個人的な内容のマイクロコンテンツ（体系化されていない短いコンテンツ）が頻繁に更新されるサイト
- ノート
ブログよりも長く、より推敲がなされたコンテンツが更新されるサイト
- フィルタ
特定の分野のニュースを取り上げ、コメントを付加する形態のサイト

と定義している。

また、Paquet は Weblog を Personal Knowledge Publishing の一形態であると捉えたうえで、これらの特徴として以下の 5 つを挙げている [22]。

1. 個人を編集主体とする (Personal editorship)

基本的には 1 人の個人が編集主体となってサイトを管理し、文書を記述する。したがって記述される情報には一定の個性が反映される。

2. ハイパーリンクによる接続構造 (Hyperlinked post structure)
1 つの文書 (エントリと呼ぶ) は短い記事からなり、参照する他のサイトの文書へのリンクを含む。また、Weblog サイトに含まれる各エントリは、Permalink と呼ばれる永続的な URI を持つ。
3. 頻繁な更新と時系列表示 (Frequent updates, displayed in reverse chronological order)
エントリは継続的かつ頻繁に記述され、公開される。新しいエントリは Weblog のメインページの最上部に掲げられ、あとは公開された順番に時系列で並べられる。
4. コンテンツへの自由なアクセス (Free, public access to the content)
課金などの制限なしに、誰でも自由に Web を介してコンテンツにアクセスできる。
5. アーカイブ形式 (Archival)
古くなった記事も削除されることなく蓄積され、Permalink によっていつでもアクセスできる。

これらは定義と呼ぶよりも、むしろ現状の Weblog サイト、とくに後述する Weblog ツールを用いて構築されたサイトの特性を示したものであるといえる。

Weblog サイトで最も頻繁に更新されるコンテンツとして、他サイトのコンテンツ紹介とそれに関するコメントが挙げられる。紹介はハイパーリンクやコンテンツ自体の引用によってなされ、その対象は通常の Web サイトやニュースサイト、他の Weblog サイトまでと多岐に渡る。Weblog サイトの書き手は、こういったサイトのコンテンツを多数紹介することで、Web 上の情報を再編集していると考えられる。その再編集は各 Weblog サイト管理者の観点によって独自になされており、その独自性によって注目されている Weblog サイトも数多い。現在では、他の Weblog サイトのコメントをさらに引用してコメントを付加するような Weblog サイトも誕生しており、これらをまとめて Weblog コミュニティと称することもある。すでにアメリカでは数十万の Weblog サイトが存在するともいわれる。Weblog は情報の受け手であった人々を、再編集という手順を通して情報の送り手に変えるという働きを持っているといえる。

すでに、Weblog サイトの数が増加するに従い、多くの関連サービスが登場している。対象を Weblog サイトに限定した検索エンジンや、引用関係を分析して最新のトピックを提示するもの、Google や Amazon.com の Web サービスと連携するものなどが挙げられる。

さまざまな分野で大きなインパクトを与えつつある Weblog であるが、その技術によってこれまでには想定されていなかったような事態も起こりつつある。既存の検索エンジンの精度の低下は大きな問題の 1 つである。多くの Weblog ツールは静的な HTML ページを多数生成す

る。しかも、これらのページは互いに密なリンク構造を有しており、代表的な検索エンジンが持つスコアリングアルゴリズムで高く評価される場合がある。その結果、検索結果の上位に Weblog サイトが並び、それらのサイトに有益な情報が存在しないといった事態が起こり得る。このように、Weblog により書くことのコストが劇的に低下したために、膨大な HTML ファイルが Web 上に流通し、すでに「情報過多」と呼ばれている現状をさらに悪化させるのではないかという懸念も見られる。

2.2.2 Weblog に関する研究

近年目覚ましい発展を遂げている Weblog であるが、これを対象とした学術的研究はまだ少数である。Weblog に関する研究の特色は、計算機上のデータとして分析・利用する計算機科学的アプローチだけでなく、Weblog の内容面に着目した社会学的あるいは心理学的なアプローチが多いことにある。これは Weblog が Web の延長線上にあるものの、よりいっそう人間のコミュニケーションに密着したものであるためであると考えられる。

計算機科学的アプローチの研究としては、Weblog によって構成される空間 (Weblogspace) が持つ性質を Web グラフの分析手法などを用いて分析した研究 [23] や、Weblog におけるトピックの伝播を抽出する研究 [24]、Weblog に適したランキングアルゴリズムの研究 [25] などがある。社会学的研究および心理学的研究には、Weblog 作者に対するインタビューに基づく Weblog の分析 [26][27] や、内容分析 (content analysis) による Weblog の分析 [28]、社会での影響力の分析 [29] などがある。なお、WWW2004^{*4}では Weblog に関する初のワークショップが開催されている [30]。

現状では、以上のように Weblog の分析が主である一方、本研究では Weblog が普及している環境でどのような支援が可能かについての検討や提案を行っている点において、これらの研究とは明確に異なる。

2.2.3 日本における Web 日記研究

日本では、Web の黎明期からいわゆる「Web 日記」を公開していた個人サイトが多く、Web コンテンツの 1 つのジャンルとして定着している。Web 日記は、その初期には手作業で HTML ページを更新・公開していたが、近年では Hyper NIKKI System^{*5}や tDiary^{*6}などの

^{*4} <http://www2004.org/>

^{*5} <http://www.h14m.org/>

^{*6} <http://www.tdiary.org/>

表 2.1 日記の分類

指向性	表現内容	
	事実	心情
自己	「備忘録」 自分のために事実を記録するもの (例: 予定などが書かれた手帳)	「(狭義の)日記」 自分のために自分の心情などを 表現するもの
関係	「日誌」 読者を意識して事実を記録するもの (例: 公開日誌, 観察日誌)	「公開日記」 読者を意識して自分の心情などを 表現するもの(例: 日記文学, 交換日記)

日記ツールや、さるさる日記^{*7}などのホスティングサービスを利用することで簡単にコンテンツを公開できるようになっている。個々のツール・サービスの詳細は文献 [31] などにまとめられている。

これらのツールにおいては RSS などのメタデータを配信する機構を持っていないものが多く、本研究のメタデータマネジメントあるいはメタデータアグリゲーションの対象にはならない。しかしながら奥村、南野らの研究では自然言語処理を用いて Web 日記のコンテンツを解析し、RSS を自動生成している [32]。この成果を利用することで膨大なコンテンツ空間を対象とすることが可能であると思われる。

ツールのレベルにおいて様々な差異はあるものの、Web 日記と Weblog は内容面でも形式面でも本質的な部分において共通であり、あえて区別する必要はない。ただ、これまでの歴史的経緯から、アメリカでは Weblog という概念の中に日記的コンテンツが含まれるのに対して、日本では日記の拡張として Weblog が位置づけられるという違いがある。

日記という私的コンテンツが Web で公開されるということについては、川浦らの一連の心理学的研究 [33][34] が明らかにしている。表 2.1 に示すように、日記は表現内容と指向性によって 4 つに分類することができる。川浦らは Web 日記も同様に 4 つの分類に広く分布することを調査によって明らかにしている。

この表によれば、日記は自分だけのために書くのではなく、他者との関係のためにも書かれるものである。逆にいえば、Web 上の日記においては、暗黙的あるいは明示的に他者とのコミュニケーションが期待されていると考えられる。Weblog や Web 日記が相互にリンクされ、Weblog コミュニティ・日記コミュニティを形成しているというのも、このような面から見れ

^{*7} <http://www.diary.ne.jp/>

ば自然な成り行きであり，Weblog ツールがさまざまなコミュニケーションの仕掛けを用意していることも Weblog にとって付随的なものではなく本質的なものであることがわかる。

なお，Web 上の個人の意見の表明の場としては，「2ちゃんねる」^{*8}をはじめとする掲示板も存在している．コミュニケーションの手段として，話題を中心に複数の書き手が集まる掲示板と，書き手が記述したコンテンツを全て1ヶ所に集める Weblog では，その特性の違いによって用途に応じた相互補完の役割を担っていると思われる [35]．

2.2.4 セマンティック Web

Web における情報洪水を克服するために，コンテンツに機械可読かつ意味理解が可能な意味的タグ（セマンティックタグと称す）を埋め込み，ソフトウェアエージェントによって適切な情報の検索や有効活用を目指す，いわゆるセマンティック Web 技術への期待が高まっている [36]．

セマンティック Web の構想は Berners-Lee によって提案されている [37]．Berners-Lee は，要素技術を階層的に並べた Semantic Web Layer Cake を順に実現することで，コンテンツの Trust，すなわち信頼性を保証することが可能になると述べている．Semantic Web Layer Cake に含まれる要素技術を以下に述べる．

- Unicode
セマンティック Web における標準の文字コードは Unicode であるとする．
- URI
セマンティック Web における全てのドキュメントや語彙は Universal Resource Identifier (URI) によって一意に識別できる必要がある．
- XML + Namespace + XML Schema
セマンティック Web におけるすべてのコンテンツおよびメタデータ語彙は，EXtensible Markup Language (XML) 記法で記述されなければならない．また，XML のドキュメント内で用いられる語彙は，曖昧性を排除するために名前空間を必要とする．
- RDF + RDFS
セマンティック Web におけるすべてのコンテンツおよび語彙は，Resource Description Framework (RDF) と呼ばれるデータモデルに則って記述される必要がある．RDF は特定のアプリケーションに依存しないリソース記述のメカニズムを提供している．RDF では全ての文 (Statement) を主語 (Subject)，述語 (Predicate) および目的語

^{*8} <http://www.2ch.net/>

(Object) の3つ組で記述する。また, RDF Schema (RDFS) は, RDF で用いるプロパティを定義するための枠組みであり, 語彙の精緻化のために利用される。

- Ontology

オントロジー (Ontology) は, さまざまな分野で定義される語彙の相互運用に必要な意味体系を指す。セマンティック Web においては, この意味体系は全て RDF, RDFS およびオントロジー記述のための上位言語である OWL[38] もしくは DAML+OIL で記述する必要がある。

- Logic

コンテンツやメタデータが正しく RDF や RDFS, オントロジー記述言語で記述されている場合, これらは推論可能な形式になっているといえる。このレイヤーでは, セマンティック Web 上のドキュメントを用いてユーザの目的に応じた推論を行うためのアルゴリズムを提供する。

- Proof

Logic 層において得られた結果が必ず証明可能でなければならないことを要請するのが Proof 層である。

- Digital Signature

コンテンツやメタデータ, 意味体系が論理的に正しいかとは別に, それ自体が不正な手段で作成あるいは利用されていないということを保証するために, デジタル署名が必要である。

この Semantic Web Layer Cake から, XML をベースとするデータ構造の統一, 意味体系の統一, 推論機構の導入というステップでセマンティック Web 環境の実現を目指すことが読み取れる。

ここに挙げた要素技術のうち, XML レイヤーのすべてと RDF および RDFS, オントロジー記述言語 OWL については World Wide Web Consortium (W3C) から正式な勧告が出されており, 構文に関する基盤は整いつつある。

しかしながら, これらの言語が HTML と同様に普及するかは未知数である。これらの言語は XML に由来する記法の複雑さを持ち, 一般のユーザが進んで記述できるようになることは考えにくい。こういった言語を専門家だけでなく一般のユーザへ普及させるためには, アノテーションやマークアップといったいわゆるオーサリング技術が必要となるが, これに関する研究は少ない。コンテンツにセマンティックタグを付加するためには, その文書のコンテキストを把握し, それに応じて適切な語彙を選択しなければならないために簡単ではない。また, タグを埋め込んだことによるメリットを即座に享受することができないといった問題もある。

さらには、既存の膨大なコンテンツについて、それらの作者に対して新たなセマンティックタグの付加を要請することは現実的に不可能である。人工知能の諸技術や自然言語処理技術による自動付加手法が検討されているが、それらが有効に機能するかどうかはいまのところ不明である。

構文レベルに関する研究が盛んになる一方で、それらの上位層にあたる Logic 層, Proof 層に関する研究は有効性を検証することが難しい。現状の研究の中心は従来の人工知能分野の成果をセマンティック Web の環境に移植することにあるが、RDF やオントロジー記述言語を利用したコンテンツがほとんど存在しないために、Web 研究で要請されるスケーラビリティの検証が不可能である。

このように、Berners-Lee が提唱するセマンティック Web の実現には障害が多く、実用的な研究を行うには長い時間がかかることが予想される。近年では、以上で述べたセマンティック Web の概念を狭義のセマンティック Web と定義し、現状の Web に存在するコンテンツを利用してより単純で実用に足る環境を構築することを目指す「Lower-case semantic web」に関する研究を行うグループも登場している。しかしながら、これを実現するには恣意的に記述された HTML からセマンティクスを抽出するための自然言語処理が必要であり、その精度を上げる段階で困難に直面している。

セマンティック Web に関する研究は数多い。現在では International World Wide Web Conference (WWW) や International Semantic Web Conference (ISWC) などの国際学会で多数の提案や検証がなされている。その中で、本研究と関連のある分野について、いくつかの研究を取り上げる。

アノテーション

メタデータのアノテーションについては、コンテンツの作り手に委ねる方法と、自然言語処理などによって自動的に行う方法に分けることができる。前者の例としては、統語・意味論などを XML 化し、コンテンツの作成の時点で構造化を行う橋田らの Global Document Annotation (GDA) が挙げられる [39]。これは、コンテンツを単純なテキストではなく意味ネットワークとして記述するためのフレームワークであり、これを利用した研究やシステムも数多い。しかしながら、このモデルでは作り手が新たに GDA の記法を理解しなければコンテンツが存在せず、したがってこれを利用した検索などが実際に利用可能になるにはかなりの時間がかかると思われる。また、過去に作成された膨大なコンテンツを GDA に対応させることは事実上不可能であり、アーカイブとしての Web の利点を享受することができない。

後者は大量のコンテンツを用い、統計的手法や自然言語処理を用いてコンテンツへのアノテーションを自動的に行う手法である [40]。この手法ではユーザの手を煩わせることがなく、

また過去のコンテンツに対するアノテーションが可能になることから，セマンティック Web 環境を実現するうえで必要不可欠であると思われる．しかし，アノテーションの正確さについては検討の余地が大きく，論理的に正確なコンテンツに基づく推論の実現というセマンティック Web の理想を実現することは難しい．

本研究では，コンテンツマネジメントシステムとの連携により，ユーザに入力負担をかけず，しかもユーザにとって明示的なアノテーションを目指している．過去のコンテンツに対しては，引用の際にメタデータを付加することを考える．これにより，小規模ながらも確実にアノテーション済みのコンテンツが増加することが期待される．

オントロジー

現在のセマンティック Web 研究においてオントロジーの構築は中心的な課題である．過去の人工知能研究では単一のオントロジーをいかに構築するかが問題とされてきたが，構築手法および維持管理の面でこのような流れは否定されつつある．問題点の詳細は文献 [41] に詳しい．代わって，対象とするドメインを限定し，詳細なオントロジーを構築する手法が主流となり，現実的な成果を得た例も見られる [42]．このようにして構築されたオントロジー同士を結合する手法やマッピングする手法についても検討が始まっており，実現可能性は高い [43]．

ただし，これらの手法についても，特定の専門家がオントロジーを定義しなければならないことには変わりなく，これらをいかにメンテナンスするのか，また統合手法についても時系列的な変化をどのように取り込んでいくかについては指針がない．

本研究では，個人がそれぞれにカテゴリ分類を行い，比較的単純な手法を用いて知識体系を組み上げるパーソナルオントロジーを提案しているが，この手法では，オントロジーに厳密さを求めないため，複数のオントロジー間の矛盾や時系列的な変化に対してもロバストであると期待される．このようなボトムアップ型のオントロジーとトップダウン型のものでは使用目的が異なると予想されるが，両者をさらに統合することによってさまざまな分野で応用できるようになる可能性もある．

Trust

セマンティック Web の目標であるコンテンツの Trust の保証に関しては，まだ研究は非常に少ない．ユーザ同士がローカルにランク付けを行い，単純な推移律によって大きなネットワークにおいてランクの自動計算を行う手法などが提案されているが [44]，人間関係に推移律を適用すること自体が適切であるかどうかについては議論がなされていない．一方，マルチエージェントや Peer to Peer (P2P) 通信の分野では，複数の動作主体の信頼性を，それぞれが持つ評価の値から計算する手法が提案されている [5]．このような研究は，いったん信頼性

を数値化できれば有効に機能するが、対象問題によって数値化の手法が大きく異なると思われるため、さらなる事例研究が必要である。

本研究では、信頼性の数値化には踏み込まず、他者の存在の可視化に徹することで、ユーザが暗黙的に理解している他者の信頼性からユーザ自身がコンテンツの信頼性を判断する補助を行う。

2.2.5 人間関係ネットワーク

本論文に含まれる第 1 の研究および第 2 の研究の双方において、人間関係ネットワークは重要な役割を果たしている。本節では、このような人間関係ネットワークの構築手法についていくつか例を挙げる。

オンラインコミュニケーションシステム

組織内の人間関係ネットワークを構築する手法として、オンラインコミュニケーションを使用し、その利用ログからネットワークを得る手法は一般的である。Adamic らは MIT および Stanford 大学の学内コミュニティシステムにおけるユーザ間ネットワークを取得し、ユーザモデルの構築に利用している [45]。また濱崎らは人工知能学会全国大会においてコミュニティシステムを構築し、人間関係ネットワークからのユーザの挙動の分析や、コンテンツの推薦を行った [46]。

これらの研究はあらかじめクローズドな環境を設定し、その中でのネットワーク構築を行っているが、本研究では Weblog という完全な開放環境を対象としている点で大きく異なる。

Tyler らの電子メールを用いた研究では大規模な組織内での複数のコミュニティの発見手法を提案している [47]。この手法はボトムアップのコミュニティ発見アルゴリズムを複数回試行し、同定された確率から各個人のコミュニティへの寄与度を測定するものである。しかしながら、この手法の目的はコミュニティのクラスタリングであり、本研究が目指すような 1 人が複数のコミュニティに所属しているといった状況を想定していない。また、Tyler らの手法ではあらかじめ大規模な組織が存在している場合でのみ適用することができる。

IKNOW は個人の持つ知識を用いて個人同士の社会的ネットワークを構築するためのシステムである [48]。IKNOW では個人を Web ページ、知識を Web ページ内のリンクとして扱い、同じリンクを持つ Web ページ同士に人間関係リンクを張る。ユーザは、結果として得られた知識ネットワークを利用して知識を検索することが可能になる。IKNOW では人間関係ネットワークが誰にでも参照可能であるとしている。一方で本研究ではパーソナルネットワークには複数のコミュニティが存在し、それぞれは競合関係にあるとの想定をしている。そのた

め、同様の社会ネットワークが構築されたとしても、本研究と IKNOW では情報の提示方法が異なる。

Web を用いた自動構築

ユーザが明示的にネットワークを構築する手法とは異なり、Web 上のデータから人間関係ネットワークを自動的に構築する手法も提案されている。Matsuo らは任意の2名の関係を検索エンジンから取得し、学習によって分類することで大規模な人間関係ネットワークを構築している [49]。

このような手法では、容易に大規模なネットワークを構築することが可能であるため、コミュニティの全体像を把握する際に適している。一方で、取得したネットワークを何らかのサービスに生かす場合には、構築したネットワークの信頼性評価が難しいため、そのまま適用するのは困難であると思われる。

本研究では、開放環境での明示的なネットワーク構築を行うため、このような問題は生じない。

ソーシャルネットワーキング

メタデータを利用した情報共有プラットフォームでは、個人とコンテンツとの関係をメタデータとして定着させることを行うが、今後は後述の FOAF メタデータ??の普及を足がかりとして、人間関係をいかに表現するかといった技術の発展が見込まれる。欧米では、近年、「ソーシャルネットワーキング」という概念のもとに Friendster^{*9}や LinkedIn^{*10}といった個人のプロフィールを収集し、それらに基づいて人材のマッチングを行うサイトが急増している。日本でも、Google による Orkut (図 2.2)^{*11}が受け入れられたのを皮切りに、GREE^{*12}, mixi^{*13}といった類似サービスが登場している。これらは、現状では単なる集中型の人間関係登録システムとなっているが、すでに Google はこれらのパーソナルネットワークを新たな検索技術に応用することを表明している。これらのサービス上で得られる情報には限界があるが、個人が持つ Weblog とのリンク付けが可能になれば、各個人についてさらなる情報を得ることができる。Weblog の効用は、個人が一定の場所において意見の表明を続けていくことで来訪者からの信頼を得やすいという部分にあると言える。上記のソーシャルネットワーキン

*9 <http://www.friendster.com/>

*10 <https://www.linkedin.com/>

*11 <http://www.orkut.com/>

*12 <http://www.gree.jp/>

*13 <http://mixi.jp/>

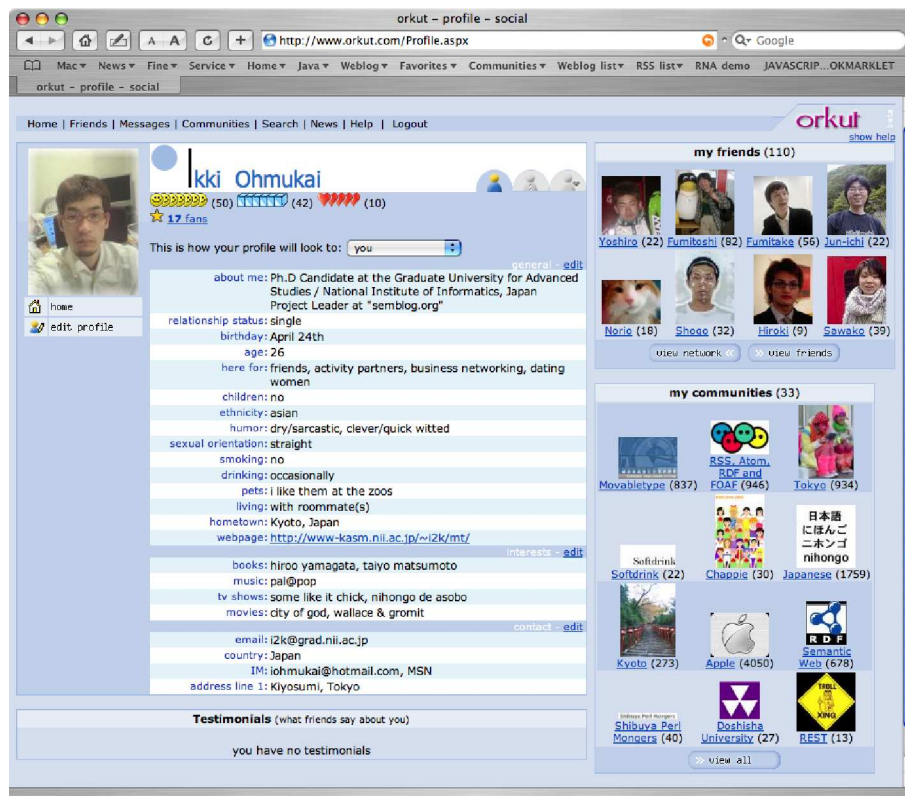


図 2.2 ソーシャルネットワーキングサービス「Orkut.com」

グ技術と組み合わせることにより、「何を」検索するといった既存の検索手法から、「誰を」検索するという Know-Who 検索を Web 上のオープンな環境で実現することが可能になる。現状のソーシャルネットワーキングサービスは、それぞれが独自のデータフォーマットを利用しており汎用性に乏しいが、今後 FOAF のようなメタデータフォーマットがインフラストラクチャーの役割を担い、サービスの統合や新しいサービスの提案につながると期待される。

2.2.6 ネットワーク分析

人間関係ネットワークの分析は，社会学的研究とコンピュータサイエンスの融合領域として注目されている．Wattsらによる Small World の研究 [50]，Barabasi によるスケールフリー構造の発見 [51] を皮切りに，これらの成果が人間関係ネットワークに適用された事例が多い．安田は，人間関係ネットワークの分析結果と個々人のパフォーマンスとの相関関係を調査した研究により [52] パーソナルネットワークの定性的な評価を行っている．

本研究では，これらの研究の知見に基づき，人間関係ネットワーク上でのコンテンツ流通を支援するためのシステムの提案と検証を行う．

2.2.7 エージェントによる情報の組織化

ソフトウェアエージェントを利用した情報の組織化については，モバイル環境での効率的な情報流通を実現するための自律的なエージェントの設計 [53] や，口コミによる情報流通を活性化させるための仲介エージェントに関する研究 [4] など，多岐に渡る．エージェントの基本的な機能や役割については，文献 [54] に詳しい．

これらの研究は，ソフトウェアがどのように人間の活動を支援するかといった視点のもとで研究されているのに対し，本研究では人間同士が直接的なコミュニケーションを取るにあたって，障害をいかに減らすかが研究の主眼となっている．このように，研究のモチベーションや目的は相反するものになっているが，技術的には共通の基盤を利用することが可能である場合が多く，両者の知見を融合することで今後の研究が進められると期待される．

第3章

パーソナルネットワークを用いた個人リソースの管理モデル

本章では、個人単位の情報流通に必要な情報のコントロール手法について議論する。まず、対象問題であるタスクスケジューリング問題のあらましを述べ、新たな解法として協調的なタスクスケジューリングというアプローチを提案する。次に、パーソナルネットワーク上のユーザ間で行われた過去のコミュニケーションの履歴情報を利用し、自動的に適切な情報のコントロールを実現する手法について検討を行う。

3.1 研究背景

本章では、Community Web モデルの実現において重要な課題となる、個人単位の情報流通に必要な情報のコントロール手法について議論する。一般に、情報を誰に送信するか、誰から受け取るかといったコントロールは、ユーザ自身が決定するものとされている。しかしながら、システム上に多数の参加者が存在し、多くの通信が発生する場合、すべての通信においてユーザが主体的にコントロールすることは現実的でない。結果として、意図しない情報の過剰な公開や、プライバシーの侵害といった問題が発生する可能性がある。そこで、本研究では、パーソナルネットワーク上のユーザ間で行われた過去のコミュニケーションの履歴情報を利用し、自動的に適切な情報のコントロールを実現する手法について検討を行う。

情報のコントロールに必要な手法は、対象とする問題によって大きく異なると思われる。そこで、本研究では、情報のコントロールがとくに必要となるような個人のタスクスケジュールリング問題を例題として選択し、これを解決するための手法を提案する。

これにより、Community Web 活動モデルの6種のプロセスの中でループが形成される Donate - Collect 間において、ユーザが意図しない情報の拡散を抑えたうえで適切な情報流通が可能になることが期待される。

3.2 タスクスケジュールリング問題

情報化社会の進展やライフスタイルの多様化は、個人と組織の関係を大きく変えつつある。現在では企業あるいは学校といった既存の組織に加えて NPO・NGO などの民間組織が多数存在し、複数の組織に同時に所属する人々はもはや珍しくない。また、これらの公的な活動組織の他に私的な活動を行うグループが無数に存在する。人々はこういった多くの組織・グループを使い分けることで各々の生活を全うしている。

このような生活の質を高めるためには、個人の持つ時間や金銭、あるいは知識といった資源（リソース）を適切に管理することが重要である。これまでに多くのリソース管理手法が提案されており、ツールやシステムとして具現化されている。これらは、組織を中心としたトップダウンモデルと、個人を中心としたボトムアップモデルに大別される。

トップダウンモデルの代表例として、グループウェアや Enterprise Resource Planning (ERP) パッケージが挙げられる [55]。これらは組織の生産性を最大限に高めるために、組織内のリソースをグループの目的に沿う形で効率よく管理、活用するためのシステムである。これらのソフトウェアは構成員のスケジュールリング支援、資源の予約や情報の共有を可能にして

おり、その利便性から多くの企業や組織で導入・活用が進んでいる。

トップダウンモデルでは、多くの組織に存在する階層関係を利用し、この階層に従って上位の人間が下位の人間のリソースを管理するを目指している。また、多くのグループウェアは組織の構成員が持つリソースを、全て組織のために利用できるという前提のもとで設計・運用されている。そのため、構成員のリソースに関する情報は全てが公開され、グループ内で管理される。しかしながら、このようなモデルは、先に述べたように、個人が複数のグループに同時に所属し、活動するといった新しいライフスタイルには適合せず、個人の希望や利益と合致しない可能性がある。

こういった問題の解決策としては、複数のグループウェアを連携する方法が考えられる。しかしながら、異なるソフトウェア間におけるデータ構造の違いやセキュリティ問題の解決といったシステムが抱える問題点が存在する。これについてはXMLの導入やシングルサインオンといった要素技術による解決が模索されているが、仮にこれらが実現されたとしても、異なる組織間での個人リソース予約の競合といった高次の問題については最終的には組織間の交渉や個人の意思決定に委ねられることになり、システムが関与できる問題ではない[56]。このため、これらの情報システムが個人にもたらす効用は極めて小さいものとなる。

一方、ボトムアップモデルの例としては手帳や Personal Information Manager (PIM)・Personal Digital Assistant (PDA) による管理手法を挙げることができる。これらの手法では個人がすべてのリソースを管理するために、複数の組織間のリソース配分は容易である。しかし、個人のための管理手法には恣意的なものが多いために効率的なリソース管理を行うことは難しい。さらには、グループ中心のアプローチとは異なり、効用を定義することが極めて難しい。そのため、従来より研究されていた人工知能や Operations Research (OR) の各手法がそのまま適用できることは少ない。筆者らはこの問題に対処すべく、意思決定論に基づくスケジューラを提案し有効性を検証したが[57]、ボトムアップモデルに共通する、管理に必要な情報をユーザが入力しなければならないコストの大きさを改善する方法については研究がなされていない。

そこで、本研究では、トップダウンモデルとボトムアップモデルの両者の問題点を解消し、主体的かつ効率的なリソース管理を可能にする協調モデルを提案する。また、このモデルを携帯電話用スケジューリングシステムとして実装し、実証実験によって有効性を検証する。

3.3 協調的スケジューリング

タスクスケジューリング問題は、個人が抱えるタスクに対して時間リソースを適切に分配する問題である[58]。基本的な制約条件としては2つ以上のタスクに同じ時間リソースを適用

できない，すなわちダブルブッキングの禁止がある．また，工場での工程管理や航空・鉄道などのダイヤ編成といった従来の工学的なスケジューリング問題とは異なり，本研究で対象とする問題では参加者である個々の人間同士，組織同士はそれぞれ独立であり，異なった目的や制約を持つものであると考えられる．そのため，これまでに研究されてきた単一目的の最適化問題や制約解消問題として扱うことは難しい．同様に，参加者どうしの情報共有の効率化による問題解決手法がナレッジマネジメントやネットワークコミュニティの分野で提案されている [54][59] が，これらは単一の組織内でのマネジメントや単一のコミュニティの形成をいかに支援するかに主眼が置かれているために，複数の人間が複数の組織を使いこなすといった状況には適用できない．

多くのタスクは個人が持つ人間関係の中からの依頼や共同作業という形で発生する．ここで人間関係とは過去のタスクの協働によって発生する2者間以上の関係であると定義する．これらのタスクは一度受理すると期日等を変更するためには相手との交渉が必要になる．とくに，こういったタスクが異なる組織によって同じ時間に複数予約された場合には，1つのタスクについて相手との交渉を行った後に，その結果をもとに別の相手との交渉をしなければならないなど，問題が複雑化し，競合解消のコストが著しく増大する可能性もある．

このような事態を回避するためには，タスクを依頼する時点で今後の交渉の余地の少ない時間帯を選択することが望ましい．

そのためには，参加者が自身の情報を公開することが重要となる．タスクの依頼者が相手のタスク状況を事前に確認し，ダブルブッキングを起こさないように配慮したリソースの予約が可能であれば受結点への到達が早まる．すでにグループウェアをはじめとするトップダウンモデルでは，組織の構成員に関する全てのタスク情報を共有することで効率的なスケジューリングを実現している．

しかしながら，この手法によって複数の組織間にまたがるスケジューリングを行う際には，プライバシーの侵害の恐れがあるために参加者に対して全ての情報の公開・共有を要請することは現実的ではない．タスクの依頼に際して必要な情報のみを適切に公開することが求められる．

このように，情報の公開に際しては，誰に，どのような情報を公開すべきかというアクセスコントロールが必要である．コントロールの手段としてはプロファイルをユーザ自身に記述させるものが多いが，この方法では入力のコストが極めて大きくなる．そこで，本研究では日常的なリソース管理の行動をシステムが分析し，その結果をもとに情報のアクセスコントロールおよびフィルタリングを実現する手法を提案する．これにより，プライバシーの保護と情報入力コストの問題を同時に解決することが可能になるとと思われる．また，ボトムアップのアプローチをとることによって，固定的ではないアドホックなグループに対する支援が可能になる．

3.4 タスクの依頼関係に基づくアクセスコントロール

本研究では、前節で述べた問題を解決するために、過去のタスクの協働関係に関する情報を利用し、新たな協働関係を結ぶための支援手法を提案する。提案システムでは、複数人の間でのタスクの依頼関係からグループの同定を行い、この結果をもとに情報のアクセスコントロールを行うことを目指す。以下にユーザモデルおよびアルゴリズムの詳細を述べる。

3.4.1 スケジューリングのプロセス

本節ではユーザ側の視点から、提案システムを用いた複数のユーザに関わるタスクの決定プロセスについて述べる。システム側の動作およびアルゴリズムの詳細については次節以降で述べる。

- ID の登録
ユーザは、提案システムを利用するために、ID とパスワードの登録を行う必要がある。
- 相互承認
提案システムによる協調的スケジューリングを行うにあたっては、事前にユーザ間の相互承認が必要となる。まず、ユーザ (A) は他のユーザ (B) に対してタスク情報の閲覧および登録を行ってよいかどうかを問うメッセージを送信する。その際には何らかのコミュニケーションを行ったうえで相手の ID を入手する必要がある。ユーザ B はこの通知に対して問い合わせを許可するメッセージを返信すると、ユーザ A とユーザ B の間に相互承認が成立したとみなされ、相手のユーザ名が関連ユーザリストに登録される。相互承認は相手 1 人につき 1 回行えばよい。
- タスク情報の閲覧
ユーザ A は、自身が持つ関連ユーザリストに含まれるユーザのスケジュール情報を閲覧することが可能である。携帯端末上のソフトウェアにおいて相手ユーザの ID を指定すると、端末はインターネット上のサーバと通信し、相手ユーザの情報を取得する。複数人の ID を指定し、情報を一括取得することも可能である。取得された情報は、ユーザ自身のスケジュール情報とともに図 3.1 に示す時系列インターフェイス上に表示される。時系列インターフェイスは上段と下段に分かれており、上段は数日間の日付が縦に並べられている。横軸は 0 時から 24 時までの時間軸である。各行の内部はユーザが関

覧したい人数分に分割されており、各ユーザのタスク情報ないし空き時間^{*1}の情報が時間軸に沿ってガントチャート形式で表示されている。この例では最上段がユーザ自身、2段目および3段目が他のユーザ情報である。他のユーザのスケジュール情報は後述のフィルタリングアルゴリズムによってコントロールされた結果のみが表示される。なお、同時に閲覧したいユーザ数が多い場合にはそれに応じて各行の高さが変わる。下段はユーザがカーソルで指定した日付に含まれるタスク情報のタイトルおよび時間が表示される。他ユーザの情報が公開されている場合にはそれについても同様に表示される。ユーザはここで表示される各タスクをカーソルで指定し、さらに詳細な情報を表示させることが可能である。

- タスクの依頼

ユーザが複数人との共同作業を行いたい場合には、まず前述の時系列インターフェイス上に相手のスケジュール情報を表示させる。次に、共通の空き時間検索機能呼び出すと、ソフトウェアは取得したデータから該当する時間帯を検索し、候補を提示する。ユーザがこの候補から適切と思われる時間帯を選択すると、図 3.3 右に示すタスク詳細設定インターフェイスが表示される。ユーザはこのインターフェイスを用いてタイトル等の情報を入力する。また、時間帯選択の時点で開始時間、終了時間およびタスクを実行するメンバー情報は自動的に入力されているが、これを手動で変更することが可能である。ただし、提示された空き時間以外の時間帯にタスクが登録された場合には警告が発せられる。全てのデータを入力し、登録すると、端末はサーバにアクセスし、他のユーザのデータに対して書き込み処理を行う。これとともに、端末内に保存されたデータを更新する。

- タスクの受理と実行

共同作業タスクがサーバに登録されると、サーバは他のユーザに対してその旨をメールにて通知する。この通知を受け取ったユーザは端末側ソフトウェアからデータ同期機能呼び出し、サーバ上のデータを取得する。その際に、新しいタスクの詳細が表示され、受理または拒否を選択する。全てのユーザが受理した場合には、そのタスクは実行可能タスクとして各ユーザのデータに正式に登録される。また、1人以上のユーザが拒否した場合にはタスクはキャンセルされ、各ユーザにその旨が通知される。タスクの依頼者は同様の作業を繰り返す必要が生じるが、その際には1度拒否された時間帯は提示されないようになる。

^{*1} 空き時間とはタスクが存在しない時間帯を示し、提案システムによって自動的に計算される。なお、夜間や休日といった、ユーザがタスクの登録を拒否したい時間帯については、これらを空のタスクとして一括登録するインターフェイスを用意し、入力支援を行っている。

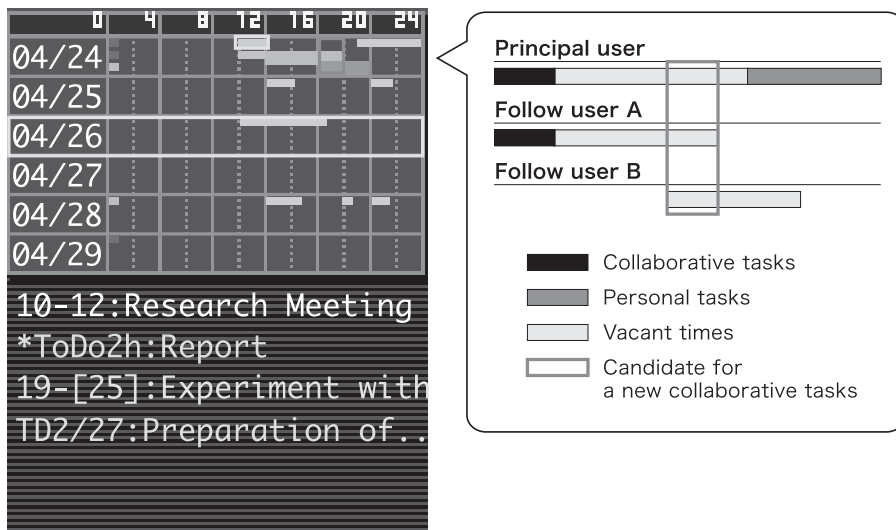


図 3.1 協調タスク表示

以上のプロセスによって登録されたタスクの一覧は図 3.2 に示すカレンダーインターフェイスで閲覧することができる。カレンダーインターフェイスではユーザ自身のタスクのみが表示される。カレンダーの各セルの下部は 3 段の簡易ガント表示になっており、タスクの登録状況を直感的に知ることが可能である。また、インターフェイスの下部は時系列インターフェイスと同様に、カーソルで指定した日付に含まれるタスクの詳細が表示される。

このように、提案システムはユーザに対して他のユーザの情報の取得、閲覧および共同作業タスクの登録作業をシームレスに支援する。ユーザは後述の人間関係ネットワークを用いた情報のアクセスコントロールといった機能の存在を意識することはない。また、事前の相互承認を除いて、こういった機能のための特別な設定は必要ない。ユーザは、日常的な行動を提案システム上で行うことにより、プライバシーを保護したうえで協調的なタスク登録を行うことが容易になる。

3.5 人間関係ネットワークを用いたアクセスコントロール

前節で述べたスケジュール決定プロセスを円滑に実行するためには、ユーザ同士が互いにタスクやスケジュールに関する情報を公開する必要がある。しかしながら、各人が異なった組織に所属しているような場合には、すべての情報の公開を要請することは適当でない。また、プライバシー保護の観点からも、互いにとって不必要な情報の公開を制限する必要がある。

一方で、共同作業に関するタスク情報のみに限定した情報共有を考えることも可能である

SUN	MON	TUE	WED	THU	FRI	SAT
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				
2003/4/24						
10-12:Research Meeting						
*ToDo2h:Report						
19-[25]:Experiment with						
TD2/27:Preparation of..						

図 3.2 カレンダー表示

が、この場合には将来の新たなタスク登録に際して情報が不足する。効果的なスケジュール決定プロセスを実現するためには、関係者から直接得られる情報だけでなく、近い関係にある第三者からの情報が重要になる。

第三者からの情報発信と受信は、1章で述べた Community Web モデルにおける Donate - Collect の関係にあたる。その際に、第三者にとってプライバシーの侵害を感じない程度まで情報公開の範囲をコントロールする手法が必要となる。本研究では、この課題を解決するために、人間関係ネットワークを用いた情報のアクセスコントロール手法の提案を行う。

提案に先立ち、重要な概念である人間関係とグループについて定義を行う。人間関係とは、タスクの依頼および協働関係に基づくユーザ間のネットワークを意味する。特定のユーザから見た人間関係は、本人を中心とするスター型の構造を形成しているが、末端である友人からは、そのノードを中心としたスター構造が別に存在するため、全体的に見ると人間関係は複雑な構造を有するネットワークとして現れる。

次に、グループの定義であるが、本研究におけるグループとは既存の組織体のように明確な境界を持つものとは限らない。ここでのグループは、共同作業に関するタスクが発生しうる人々の小集合であり、人間関係ネットワークの部分ネットワークとして定義される。提案手法においては、このグループは過去の協働関係に応じて動的に決定される。

提案手法では、情報アクセスコントロールは、このグループの単位で実行される。これは、

TYPE	Collabo. Schedule
TITLE	Research meeting
FROM	i2k
MEMO	15F seminar room
MEMBR	i2k:ham:akatsuka
START	2003/4/24 10:00
LOAD	2 hours
END	2003/4/24 12:00
PRIOR	MidHi PROG. 30%
PUB.	Smart REPT. 2 weeks

図 3.3 タスク詳細設定

共同作業タスクが存在した場合、そのタスクが存在するグループのメンバーには情報が公開され、そうでないユーザには公開されないことを意味する。

タスクによっては、タスクの関係者の数がグループの構成員の数と一致する場合もあれば、グループの構成員の方が多くなる場合もある。前者におけるアクセスコントロールの結果は、タスクの関係者のみで情報を共有する場合と何ら変わらない。一方、後者では、そのタスクに直接は関係はないがグループには含まれているユーザ、すなわち 3.4 におけるユーザ C に対しても情報が公開されることになる。このユーザ C は、過去にユーザ A および B との協働関係があったために同一のグループにみなされたのであり、過去に協働関係のないユーザには情報は公開されない。

このように、情報の公開範囲を適切に拡張する手法を導入することで、円滑な意思決定が可能になると期待される。本手法は、ユーザが所属するグループを自動的に発見するアルゴリズムと、発見された複数のグループから適切にアクセスコントロールを行うアルゴリズムから構成される。詳細を以下に述べる。

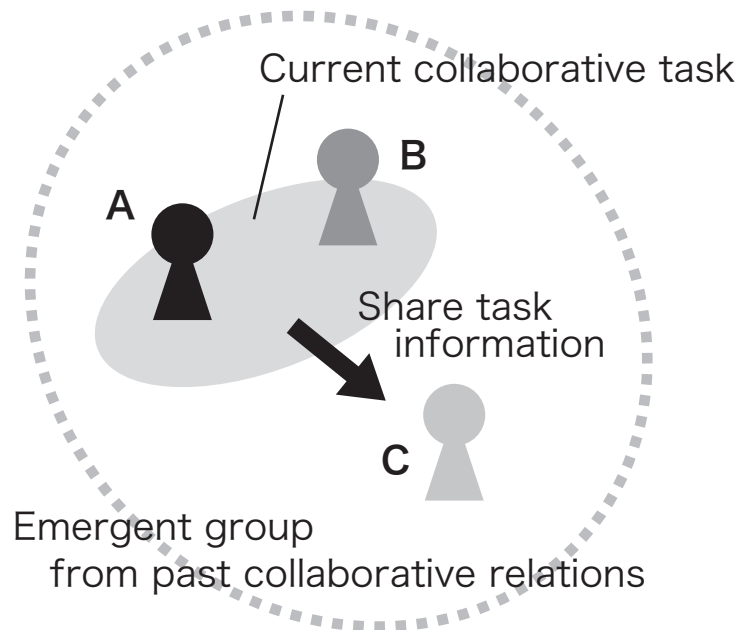


図 3.4 グループの定義と情報共有範囲の拡張

3.5.1 人間関係ネットワークからのグループ発見

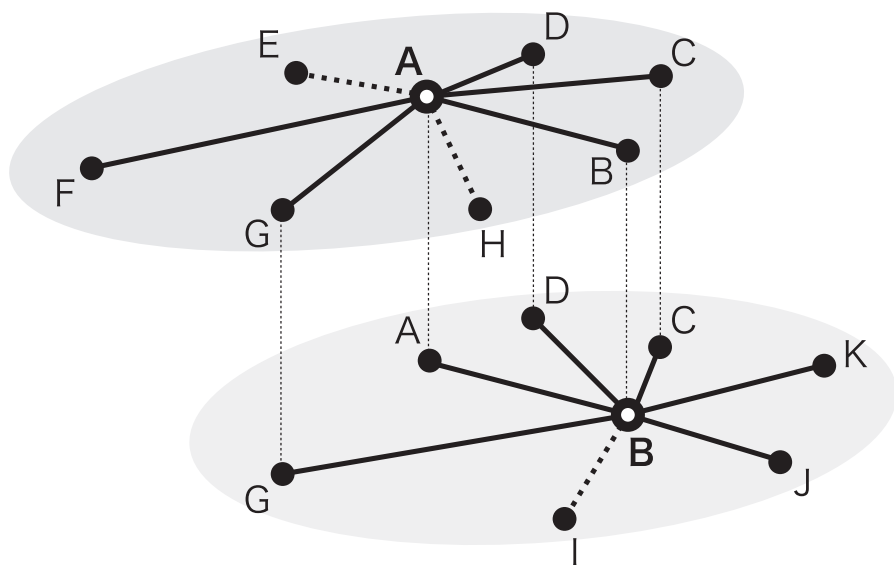
本研究が対象とするような状況においては、ユーザが所属するグループの構成は動的に変化する。また、時間の進行にしたがって新たなグループが生起する可能性がある。このような場合、ユーザによって明示的にグループを設定させることは、ユーザにとって大きな負担になるだけでなく、設定が不正確になり得る。そこで、本研究ではグループを自動的に発見する手法を提案する。

この手法では、過去のタスクの依頼および受理関係に基づく人間関係ネットワークを形成し、その中で有意な部分を抽出して、これをグループとして用いる。ここでは有意な部分とは人間関係ネットワークをグラフとしてみた場合の極大完全部分グラフである。

ユーザ間での相互承認およびタスクの依頼といったインタラクションが発生すると、システムはそれらのユーザ同士を人間関係ネットワークに登録する。人間関係ネットワークはユーザごとに構築されるものとし、それぞれは図 3.5 に示すようにユーザ本人を中心としたスター型のトポロジーとなる。また、各リンクには「Authorized (相互承認済み)」もしくは「Collaboration (タスク依頼関係あり)」のラベルが付加される。

あるユーザ (*self*) が他のユーザ (*target*) のスケジュール情報を閲覧する際に、システム

Personal Network of A



Personal Network of B

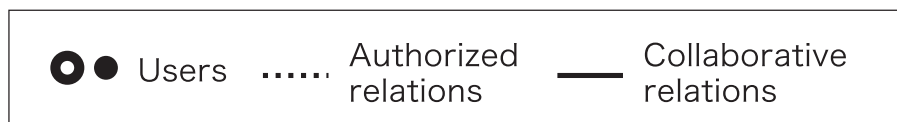


図 3.5 ユーザごとの人間関係ネットワーク

はこれらの人間関係ネットワークから両者がともに所属しているグループを発見する．グループ発見アルゴリズムの手順を図 3.6 に示す．まず，*self* と *target* の 2 名を最小のグループとして登録する．次に，*self* がパスを持つ任意の 1 ユーザ (*candidate*) を選択し，グループ内の各ユーザに対して *candidate* へのパスが存在するかどうかを問い合わせる．全てのユーザがパスを持っていた場合には，*candidate* をグループに追加し，このグループをリストに登録する．登録の際には，すでにリストに存在するグループと構成メンバーを比較し，前者が後者を包含する場合には後者をリストから削除する．これらの操作を再帰的に繰り返し，最終結果をグループリストとして出力する．なお，*self* が複数人のデータを取得する際には，それぞれのユーザごとに複数回繰り返す．

本手法の計算量は，*self* と *target* の両者がともにパスを張っているユーザ数 N の増加にしたがって 2^N のオーダーで増加する． N の最大値は，*self* および *target* が持つパスの数をそれぞれ n_{self} ， n_{target} とすると， $\min[n_{self}, n_{target}]$ となるが，一般的に，1 人が持つパスの数

は、システム全体のユーザ数には比例せず、一定の数で飽和する*²。よって、計算量が全体のユーザ規模にしたがって無限に増加することはない。また、同定されるグラフの規模が比較的小さいと予想されることから、深さ優先の探索手法を取り入れ、早い段階での枝刈りを可能にしている。しかしながら、大規模な実運用に際しては、極大完全部分グラフの探索の高速化手法 [61] などを取り入れる必要があると思われる。

3.5.2 グループによるアクセスコントロール

次に、上で求めたグループを用いて、ユーザ *self* によるユーザ *target* のスケジュール閲覧の可否を決定する。この場合、スケジュール閲覧の可否とはユーザの持つ個々のタスクの詳細を表示するか否かを指す。

前節の手法により、*self* および *target* を含むグループ集合 *FoundGroups* を得ることができる。ユーザの環境によっては複数のグループが発見される可能性がある。次に、システムは *target* の持つ全てのタスクより *self* に公開可能なものを選択する。選択アルゴリズムを図 3.7 に示す。

いま *self* をユーザ *A*、*target* をユーザ *B* とした場合に発見されたグループを $G_1 = \{A, B, C, D\}$ 、 $G_2 = \{A, B, G\}$ とする。

あるタスク T_1 に含まれる共同作業者のリストを $C_1 = \{B, G\}$ とする。このタスクについて $C_1 \subseteq G_2$ が成立するため、タスク T_1 はユーザ *A* および *B* がともに所属するグループ内で行われるタスクであるとして *A* に公開される (図 3.8(a))。一方、 $C_2 = \{B, K\}$ であるようなタスク T_2 においては $C_2 \not\subseteq G_1, G_2$ であるために公開されない (図 3.8(b))。

以上で述べた手法により、過去の協働関係からのユーザ間のネットワークの構築およびグループの発見、グループ内のアクセスコントロールを自動的に行うことが可能になる。ユーザはグループの構成員をあらかじめ設定するなどのプロファイリングの必要がない。他人へのタスクの依頼を行う操作のみで複数のグループ間にまたがる情報のアクセスコントロールが実現される。

なお、ユーザは個々のタスクに対し、この手法を利用したアクセスコントロールを適用する、全てのユーザに公開する、全てのユーザに公開しない、という3段階の情報公開レベルを設定することができる。

*² 社会学の分野での調査 [60] によれば、1人が持つ知人の数は平均40人程度であるとの報告がある。

3.6 本章のまとめ

本章では、Community Web 活動モデルの適用事例としてタスクスケジューリング問題を取り上げ、これを概説した。そして、複数人による協調モデルによる解法を提案し、理想的なユーザモデルについて述べた。ただし、実世界では複数の組織に所属する個人間での情報共有はプライバシーの侵害の恐れがあるため、容易ではない。そこで、本研究では過去の協働の履歴から人間関係ネットワークを構築し、ネットワーク分析の結果から共有すべき情報のフィルタリングを行う手法を提案した。

次章では、これまでに述べてきたアクセスコントロール手法を携帯電話上のアプリケーションとして実装し、実証実験を行うことで有効性を検証する。

```
procedure FindMaximalPartialGraph(self, target)
begin
   $g \leftarrow \{self, target\}$ 
   $FoundGroups \leftarrow \{g\}$ 
   $cl \leftarrow \langle self \text{ が持つ Collaboration リンクのユーザーリスト (target を除く)} \rangle$ 
  for each  $x \in cl$  do
    FindPartialGraph( $g, cl - x, x$ )
   $\langle FoundGroups \text{ を出力} \rangle$ 
end

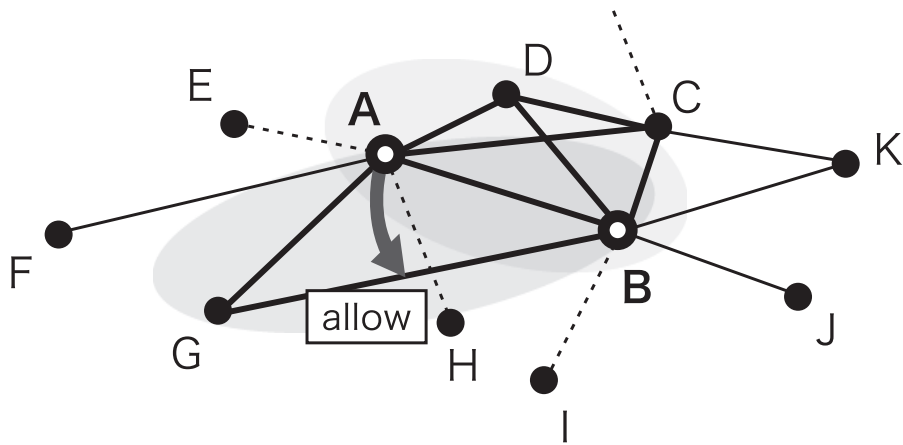
procedure FindPartialGraph(group, clist, candidate)
begin
  if  $\langle group \text{ の全ての要素が } candidate \text{ への Collaboration リンクを持つ} \rangle$  then
    begin
       $\langle candidate \text{ を } group \text{ に追加} \rangle$ 
      AddGroup(group)
      for each  $x \in clist$  do
        FindPartialGraph(group,  $clist - x, x$ )
    end
  end
end

procedure AddGroup(group)
begin
  for each  $g \in FoundGroups$  do
    begin
      if  $group \supset g$  then  $\langle g \text{ を } FoundGroups \text{ から削除} \rangle$ 
      if  $group \subseteq g$  then exit
    end
   $\langle group \text{ を } FoundGroups \text{ に追加} \rangle$ 
end
```

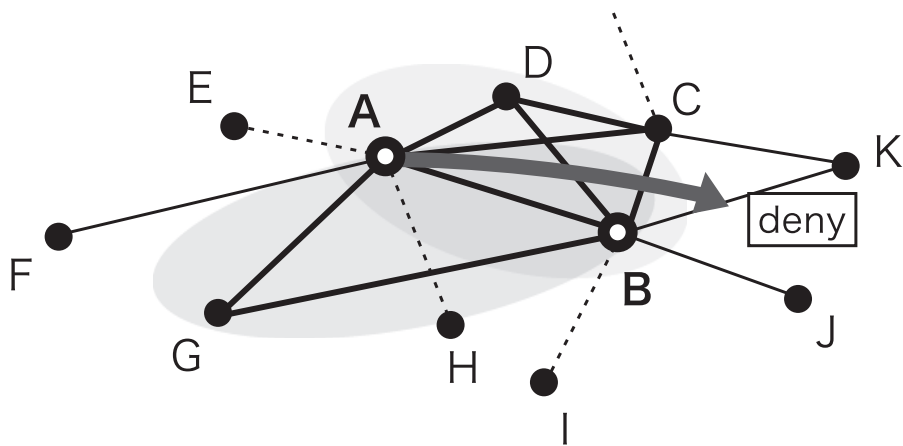
図 3.6 グループ発見アルゴリズム

```
procedure FilterTask(self, target, FoundGroups)  
begin  
  tasklist ← <target が持つタスクのリスト>  
  for each task ∈ tasklist do  
    begin  
      for each group ∈ FoundGroups do  
        begin  
          if <group のメンバーリストが task の共同作業リストを包含する> then  
            <task を self に公開>  
          end  
        end  
      end  
    end  
  end  
end
```

図 3.7 アクセスコントロールのアルゴリズム



(a) Acceptable access in the working group



(b) Unacceptable access

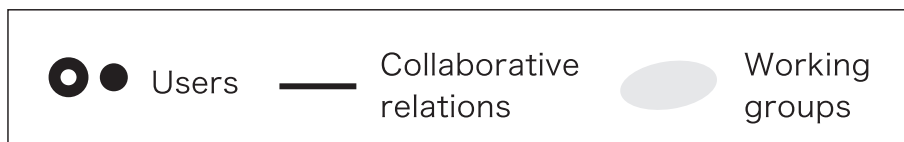


図 3.8 グループ発見を用いたタスク情報のアクセスコントロール

第4章

協調的タスクスケジューラの実装と評価

本章では、Community Web モデルの実現にとって必要な、提案手法の実用化について議論を行う。前章で述べた情報のアクセスコントロール手法を一般のユーザに普及させるには、アルゴリズムの妥当性だけでなく、それを利用することがユーザにとって快適なものであるかどうかや、データ構造の適切さ、インターフェイスの適切さなどについて検証を行う必要がある。また、アルゴリズムについても、多くのユーザが利用する際のスケーラビリティを考慮する必要がある。

そこで、本研究では、個人間のタスクスケジューリング支援手法を、携帯電話上のアプリケーション「Social Scheduler」として実装し、これを用いて実証実験を行った。そして、この実験から得られるパーソナルネットワークと、提案手法による情報のアクセスコントロールに関する性能との関連について議論する。

4.1 携帯電話への実装

個人の日常的なスケジューリングを支援するためには、利用シーンに限られる PC よりも、常に持ち歩くデバイスが適していると考えられる。また、複数人での協調的なタスク登録作業を支援するためには通信機能が必須である。そこで、本研究では提案手法を携帯電話上で動作するソフトウェアを用いたサーバ・クライアントシステムとして実装した。これにより、全てのユーザが場所および時間を問わず互いのスケジュール情報を閲覧し、タスクの依頼を行うことができるようになる。

一方で、クライアントに携帯電話を利用することによる制約も大きい。現状の携帯電話環境では端末同士の直接通信は不可能であるため、インターネット上のサーバに各ユーザのタスク情報を格納している。同様の情報は各端末内にも保存されており、データの同期はユーザが明示的に指定した時点で行われる。相互承認ネットワークやタスクの依頼関係ネットワークはサーバ上でのみ保存される。また、端末へのプッシュ型通信が行えないことから、他のユーザが登録した共同作業タスクの存在を知るための時間差が生じる。そこで、提案システムでは携帯電話のメール機能と連携し、サーバがタスクを登録した時点で共同作業に対してその旨を伝えるメールを送信する。メールにはタスク情報の概要が記載されているほか、埋め込まれたリンクからクライアントソフトウェアを起動することができる。

インターフェースの制約としては、表示画面の小ささや解像度の低さが挙げられる。情報の一覧性の低さはこれまでに述べた協調的なタスク登録プロセスを妨げる原因となるため、既存の携帯電話用スケジューラと比較して 1 画面中の情報量を高められるよう、カレンダーインターフェイスおよび時系列インターフェイスを設計している。また、携帯電話の機種によって解像度が異なる問題に対しては、機種ごとのプロファイルを作成し、ソフトウェアのダウンロード時に機種を判別して適切なプロファイルを送信する。

4.1.1 システム構成

Social Scheduler のシステム構成を図 4.1 に示す。

Social Scheduler はクライアント・サーバ型のシステムである。インターネット上のサーバと携帯電話間においては TCP/IP 等を用いた通信が不可能であることから、株式会社 NTT ドコモが提供する通信センター (i モードセンター) を介し、サーバ-i モードセンターでは TCP/IP、i モードセンター-携帯電話間では独自プロトコルを利用している。アプリケーションプロトコルとしては HTTP を利用している。HTTP は両通信プロトコル上でシームレス

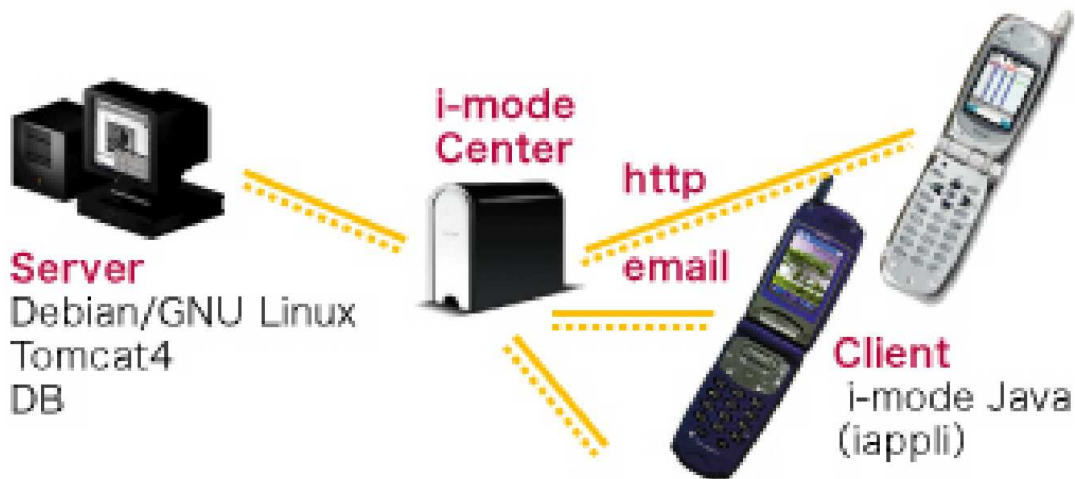


図 4.1 システム構成

に動作するため、システムの開発にあたっては通信プロトコルの差異を考慮する必要はない。また、操作によっては E-mail による通信を利用している。現在の携帯電話システムにおける E-mail はインターネットメールと互換性があるため、こちらもシステム間の差異を考える必要はない。

各クライアントはインターネット接続が可能な携帯電話である。これらの端末は Java アプリケーションのダウンロードおよび実行が可能である。本研究では株式会社 NTT ドコモが提供する端末（ソニー株式会社製 SO504i、日本電気株式会社製 N504i、N504iS、および松下電器産業製 P504iS）を利用した。ソフトウェア開発環境としてはサン・マイクロシステムズ株式会社が提供する Java 2 Micro Edition Connected Limited Device Configuration (J2ME CLDC) [62] および J2ME Wireless SDK for DoJa[63] を利用した。

サーバは常時インターネットに接続されている。ハードウェアは一般的な PC サーバを用いた。オペレーティングシステムには Debian/GNU Linux 3.0 を使用し、Java によるアプリケーションサーバ実行環境としてサン・マイクロシステムズ株式会社が提供する Java 2 Enterprise Edition (J2EE) 1.2、Java 2 Standard Edition Runtime Environment (J2SE JRE) 1.4、および Apache Project が提供する Tomcat 4.0 を利用した。

4.1.2 データ構造

すべてのタスク情報はサーバ上のデータベースと携帯端末内に格納されている。両者はユーザが明示的に指定することで同期をはかることが可能である。本システムではタスクを個人タスクと協調・依頼タスクの2種類に分類している。個人タスクはそれを保持するユーザ自身のみ関係のあるタスクであり、協調・依頼タスクは複数のユーザに関係するタスクである。後者のうち、協調タスクとはタスクの実行者の中に依頼者を含むものであり、依頼タスクとは依頼者を含まないものである。あるユーザによって協調・依頼タスクが発行されると、サーバはタスク情報からそのタスクに関連のあるユーザリストを取得し、それぞれの個人タスクデータベースに新たな個人タスクとして追加する。協調・依頼タスクを依頼されたユーザが同期を行うと、サーバ上の個人タスクデータベースから携帯端末に協調タスクの情報がダウンロードされ、タスクの依頼が通知される。

詳細なデータ構造は付録に示す。

4.1.3 クライアントの実装

クライアントソフトウェアは主にカレンダービュー、タスクの登録・変更画面、協調タスクビュー、およびユーザマネージャの4種のインターフェイスから構成されている。本節ではそれぞれのインターフェイスが備える機能について述べる。また、それぞれのインターフェイスに付随するメニュー項目についても述べる。

カレンダービュー

カレンダービューはクライアントソフトウェアのデフォルト画面として起動処理の終了後に表示される。カレンダービューの例を図4.2に、実機における表示例を図4.3に示す。カレンダービューは携帯端末の小さな画面に合わせて様々な情報を効率よく表示することを目指して設計されている。表示は上部と下部に大きく分けられ、上部では1ヶ月分のカレンダーと各日付のタスクの有無を表示する。日付の下は6~24時を表す時間軸を3段に分割し、スケジュールタスクが存在する時間に合わせて棒状に表示する。日付の横にはTODOタスクの有無を表示し、赤はTODOタスクの締切が存在する日、青はTODOタスクを行わなければならないと予想される日を表している。また、紫は本日であることを表している。カーソルキーで上部のカレンダー上のフォーカスを変えることができる。フォーカスが最上段にある場合にさらに を押すと前月に、フォーカスが最下段にある場合にさらに を押すと次月に移動する。ま

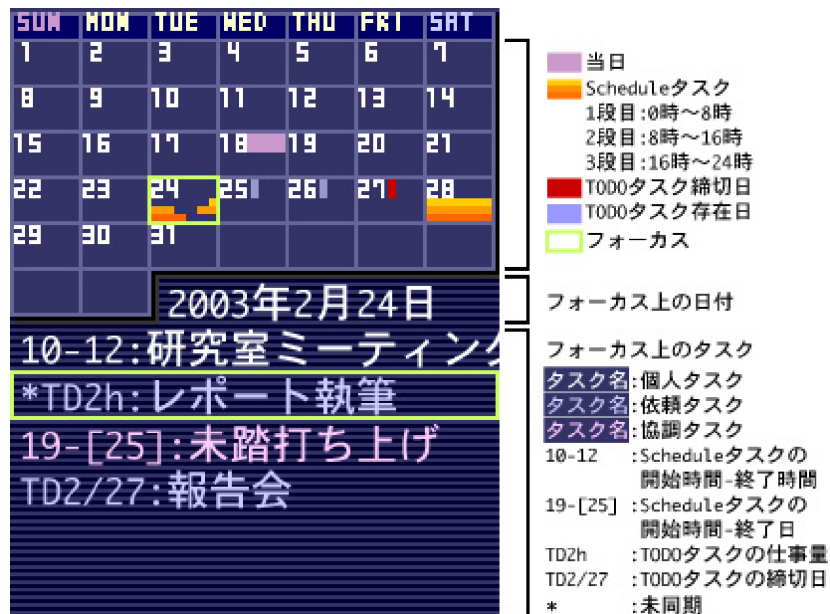


図 4.2 カレンダービュー

た、数字キーの「2」「4」を押すと前月に、「6」「8」を押すと次月に移動することも可能である。「5」を押すとフォーカスが本日に戻る。フォーカスされた日付にタスクが存在する場合には、画面下部にタスク名と時間が表示される。ここでは文字の色とプレフィックスによってタスクの種類を判別することが可能である。白文字のタスクは個人タスク、青文字のタスクは依頼タスク、赤文字のタスクは協調タスクを示す。また、タスク名の前に「TD」と記載されたタスクは TODO タスク、その他のタスクは Schedule タスクを示す。タスク名の前に「」が記載されたタスクはサーバとの同期がなされていないタスクであることを示す。フォーカスの日付にタスクが1つのみ存在する場合には決定キー、2つ以上の場合にはカーソルでタスクを選択して決定キーを押すとタスクの詳細データ（後述）を見ることができる。



図 4.3 カレンダービュー（実機）

タスクの登録・更新画面

タスクの登録・更新画面では新しいタスクのデータ入力および既存のタスク情報の変更を行うことができる。本画面ではタスクに関する全てのパラメータを一覧することが可能であり、これをもとにユーザは個々のタスクに関してパラメータの設定を行う。タスクの登録・更新画面の例を図 4.5 に示す。各タスクに設定できるパラメータと表示との対応は以下の通りである。

- TYPE：入力項目 (A)(B)

タスクの種類を設定する。個人・協調・依頼の 3 種類 (A) と Schedule・TODO の 2 種類 (B) の組み合わせで計 6 種類のタスクが設定可能である。

- 個人

個人で行うタスク、もしくは ngrid を利用していないメンバーとの協調作業を行う場合に設定する。

- 協調

ngrid を利用しているメンバーとの協調作業を行う場合に設定する。

- 依頼

ngrid を利用しているメンバーにタスクを依頼する場合に設定する。

- Schedule

開始時間と終了時間が決定されており、その間は拘束されるようなタスクである場合に設定する。例：会議・授業

- TODO

締切時間のみが決定されており、実際のタスクの実行時間がユーザの裁量に任せられるような場合に設定する。例：論文執筆

- TITLE：入力項目 (C)

タスクの名称

- FROM：入力項目 (D)

タスクの依頼者。基本的には手動で入力を行うが、協調タスクを登録する場合は自動的に入力される。

- MEMO：入力項目 (E)

タスクに関する備考

- MEMBR (Member の略)：入力項目 (F)

協調タスクを行うメンバーの ID。個人タスクの場合は空欄 (図 4.7)

- START：入力項目 (G)

Schedule タスクの開始日・時刻

- LOAD (Workload の略) : 入力項目 (H)(I)

TODO タスクの仕事量の見積もり。単位は時間・日・週・月から選択する。

- END : 入力項目 (J)

Schedule タスクの終了日・時刻および TODO タスクの締切日・時刻

- PRIOR (Priority) : 入力項目 (K)

タスクの優先度。High / Mid-High / Middle / Mid-Low / Low の 5 段階。主観的評価を行い入力する。

- PROG. (Progress) : 入力項目 (L)

TODO タスクの進捗度。0 ~ 100% の 10 段階

- PUB (Publication) : 入力項目 (M)

タスク情報の公開・非公開。

- all : 認証関係を結んだ全ての人に公開
- smart : グループ内のメンバーにのみ公開 (詳細は 3 . 4 . 2 節で述べる)
- off : 公開しない

- REPT. (Repeat) : 入力項目 (N)(O)

定期的タスクの一括入力支援機能。繰り返し回数を数字で入力し、毎日・毎週・毎月を選択する。

- off : 繰り返ししない (1 回みのタスク)
- days : 毎日繰り返す
- weeks : 毎週の同じ曜日に繰り返す
- month : 毎月の同じ日に繰り返す
- mon_w : 毎月の同じ週・曜日に繰り返す



図 4.4 新規タスク



図 4.5 新規タスク (実機)



図 4.6 文字入力画面（実機）



図 4.7 協調メンバーの変更

協調タスクビュー

協調タスクビューでは各タスクを 1 日ごとのガントチャート（クロノグラム）上に配置する。また、認証関係を持つユーザのスケジュール情報を取り込み、自身のスケジュールと重ね合わせて表示することが可能である。協調タスクビューの例を図 4.8 に示す。最上段に表示されたオレンジ色のバーがユーザ自身のタスクである。カーソルキーでフォーカスを移動させ、選択すると各タスクの詳細が表示される。日付の横に表示された赤の正方形は TODO タスクが存在することを示している。10 時から 12 時までのタスクの直下に同じ長さの紫色のバーが存在する。これは認証関係を持つ他ユーザ A が公開しているタスクである。紫色はこのタスクが協調タスクであることを示している。この協調タスクの右に表示された青いバーは、ユーザ A がこの時間帯にタスクを保持していない（空き時間である）ことを示している。また、その下の段に存在する青いバーは他ユーザ B の空き時間である。これによれば、16 時から 18 時にかけてはユーザ自身にタスクが存在せず、かつユーザ A と B の空き時間が存在する。このような場合、その時間帯に 3 人が関係する協調タスクを投入すると受理される可能性が最も高くなる。そこで、本ソフトウェアでは登録されたデータより、協調ユーザに指定した全てのメンバーに共通する空き時間を自動的に算出し、紫色のフォーカスとして提示する。このフォーカス上で選択を行うと、タスク登録画面に移動する。この場合、算出された時間とメンバーリストがすでに記載された状態になる。

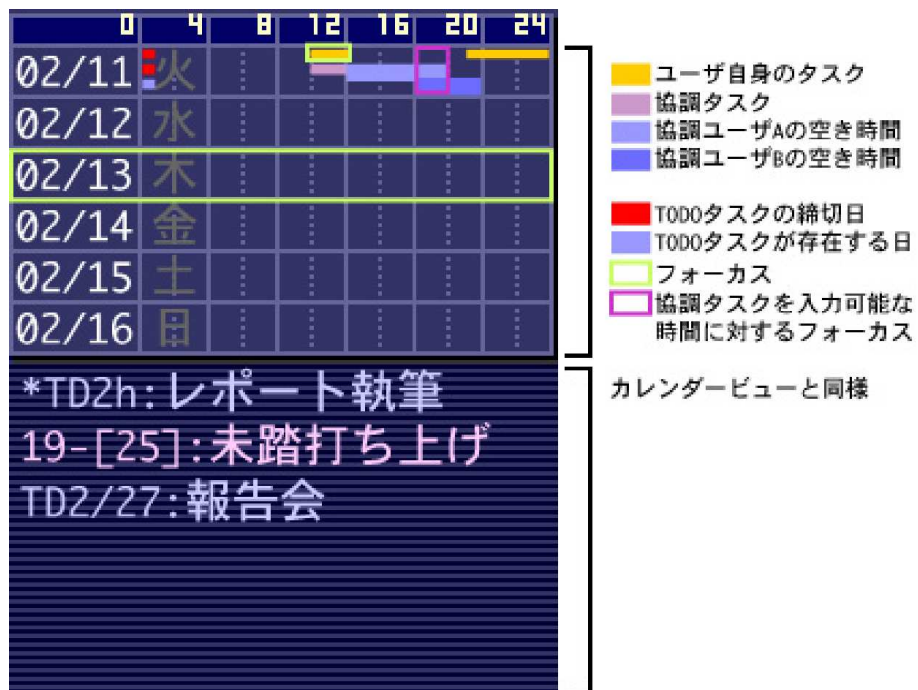


図 4.8 協調タスクビュー



図 4.9 協調タスクビュー（実機）



図 4.10 協調タスクビュー（実機）

ユーザ名	Out	In	共有/全グループ	評価値
USER	→	←	GROUP	EVAL.
akatsuka	3	8	4/11	.384
ham	7	6	7/11	.631
coba	2	0	1/11	.107
ikns	1	3	2/11	.189

認証ユーザリスト
 フォーカス

*TD2h: レポート執筆
 [19]-[25]: 未踏打ち上げ
 TD2/27: 報告会

タスクリスト

図 4.11 ユーザマネージャ

ユーザマネージャ

ユーザマネージャではこれまでに認証関係を結んだ全てのメンバーリストを表示する。また、ユーザごとのタスク依頼数（Out と表記）、タスク被依頼数（In と表記）、グループ共有数、ユーザ自身が保持するグループ数が表示される。表示はこれらの値をキーにソートし直すことが可能である。また、各ユーザにフォーカスを合わせると、そのユーザに関係したタスクの一覧が画面下部に表示される。ユーザ自身にフォーカスを合わせた場合には、全タスクが表示される。このタスクリストでは、フィルタ設定により表示条件を変更することが可能である。

メニュー

カレンダービュー、協調タスクビュー、およびユーザマネージャでは左ソフトウェアキーから基本メニューを呼び出すことができる。メニュー項目の内容を以下に述べる。

- 新規タスク

タスク登録画面に移動する。タスクの START パラメータはカレンダービューおよび協調タスクビューではフォーカスされた日付に設定される。ユーザマネージャから呼び出された場合には本日の日付になる。

- 協調メンバー設定

協調タスクビューにおいて公開スケジュールを閲覧したいメンバーを指定する。

- 同期

同期には通常の同期と完全同期の 2 種類がある。

- 同期

同期が行われていない携帯端末内のタスク情報およびサーバに保存された情報を同期する。その際、他のメンバーによって登録された協調タスク・依頼タスクの情報がサーバに存在すると確認を促すダイアログが表示される（図 4.14）

- 完全同期

同期が行われている、いないに関わらず、携帯端末内のタスク情報とサーバに保存されたタスク情報一致させる。通常の同期と比較して通信量が多い。主にソフトウェアのアップデートの際に利用する。

- 環境設定

環境設定には使用頻度の少ない設定項目が収められている。

- 協調ユーザリスト取得

新たに他ユーザと認証関係を結んだ場合に、携帯端末内に保存された認証ユーザリストを更新する。

- 協調ユーザ追加

他ユーザと新たに認証関係を結ぶ場合に相手の ID を入力する。

- ユーザ設定

別の ID に変更する場合に利用する。

カレンダービュー、協調タスクビュー、およびユーザマネージャでは右ソフトウェアキーから表示メニューを呼び出すことができる。表示メニューでは、これらの画面表示を切り替えることが可能である。右ソフトウェアキーを 2 回続けて押すと、カレンダービューと協調タスクビューのトグル切替が可能である。



図 4.12 メニュー（実機）



図 4.13 メニュー（実機）

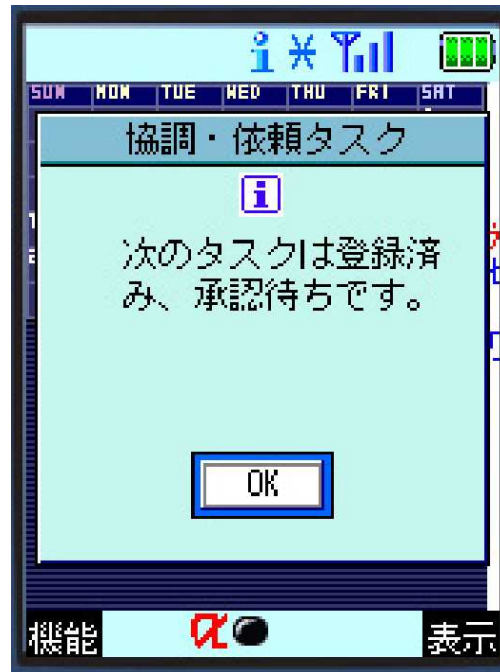


図 4.14 同期後の確認ダイアログ (実機)



図 4.15 同期後の確認ダイアログ (実機)



図 4.16 環境設定 (実機)

4.1.4 サーバの実装

サーバソフトウェアはデータベースモジュールおよびプログラムモジュールから構成されている。本節ではこれらの詳細について述べる。

データベース

データベースモジュールは以下のデータを格納している。

- ユーザ情報 (ユーザ DB)
すべての登録ユーザの ID, パスワード, および携帯電話のメールアドレスを格納する。
- ユーザごとの認証ユーザリスト (認証ユーザ DB)
認証関係を結んだユーザの一覧を保持する。このデータベースは全ユーザの数だけ用意される。
- ユーザごとのタスク情報 (タスク DB)
各ユーザが持つすべてのタスクとそのパラメータを保持する。このデータベースは全ユーザの数だけ用意される。
- 協調タスク情報 (協調タスク DB)
協調タスクおよび依頼タスクに関する情報は、ユーザごとのタスク情報とは別にこのデータベースに集めて一括管理する。

プログラムモジュール

サーバは以下に述べる 8 種のプログラムモジュールから構成されている。

- ユーザ登録モジュール
本システムを初めて利用する際に ID, パスワードおよび携帯端末に付加されたメールアドレスの入力を要求する。これらのデータはユーザ DB に格納する。登録が正常に行われると、携帯端末にメールを送信する。
- ダウンロードモジュール
クライアントソフトウェアのダウンロードページにアクセスした携帯端末の機種およびモデルを判別し、それぞれに適したアプリケーション記述ファイルを配信する。各端末はこのアプリケーション記述ファイルに沿って適切なソフトウェアを自動的にダウンロードする。

- 認証モジュール
クライアントソフトウェアからの協調ユーザ追加リクエストを処理する。リクエストの発信元の ID と相手の ID からそれぞれの認証ユーザ DB を検索し、認証関係が結ばれていなければ相手の携帯端末に特殊な URL が記載されたメールを送信する。相手がこのメールを確認および承認し、URL に接続すると認証モジュールがこれを感知して各認証ユーザ DB にそれぞれの ID を追加する。また、クライアントソフトウェアから協調ユーザリスト取得リクエストがあった場合には、認証ユーザ DB の内容を送信する。
- 同期モジュール
クライアントソフトウェアの同期リクエストを受け取り、携帯端末内のデータとサーバ上のタスク DB を同期させる。通常の同期の場合、まず両者のタスク情報からタスク ID を検索し、どちらかに存在しないタスクがあればこれをコピーする。両者に存在してもタイムスタンプが異なる場合には新しいものを古いものに上書きする。最後に両者のタスク情報に付加された同期フラグを判別し、未同期であると見なされた場合にはすべてのパラメータをチェックし、問題があればこれを修正する。最後に全てのタスクの同期フラグを同期済みに設定する。完全同期のリクエストがあった場合には、同期フラグのチェックを行わず、全てのタスクについてパラメータのチェックと同期および修正を行う。協調タスクもしくは依頼タスクが新たに登録されるかパラメータが変更されていた場合には協調タスクモジュールを呼び出す。
- 協調タスクモジュール
同期モジュールより協調タスクもしくは依頼タスクの新規登録が伝えられた場合には、このタスクの内容を協調タスク DB に登録する。また、このタスクに関連するメンバー全員に承認するかどうかを決めるためのメッセージを送信する。このメッセージに対する反応があった場合、協調タスク DB 内のパラメータを随時書き換え、全てのメンバーが承認すると依頼者に対して登録完了のメッセージを送信する。パラメータに変更があった場合にも同様の作業を繰り返す。
- データアクセスモジュール
クライアントソフトウェアの協調タスクビューにおいて他ユーザの情報閲覧リクエストが発せられると、認証ユーザ DB をチェックし、許可および不許可を判別する。許可された場合、データアクセスコントロールモジュールを呼び出す。同じくクライアントソフトウェアのユーザマネージャからユーザ情報取得リクエストを受けた場合には、タスク DB および協調タスク DB を検索し、必要な情報を送信する。また、データアクセスコントロールモジュールおよび PageRank モジュールを呼びだして必要な情報を受け取り、これをクライアントソフトウェアに送信する。

- データアクセスコントロールモジュール

データアクセスコントロールモジュールはあるユーザが他のユーザのタスク情報を閲覧してよいかどうかを判定する。

4.2 実証実験

提案システムが適切に情報のアクセスコントロールを行えるかを確認するために、実証実験を行った。期間は2003年1月26日から2月17日までの約3週間である。被験者は16名である。実験内容は、期間中にクライアントソフトウェアを実際に利用し、被験者自身のタスクを入力してもらうというものである。なお、被験者のグループ構成は、A研究グループに属するのが3名、B研究グループに属するのが2名、C研究グループに属するのが5名、C研究グループのOB(Dグループとする)が3名、それら以外に個人的な友人関係が含まれる。

4.2.1 実験の手順

実験中に被験者が行った作業を以下に述べる。まず、提案システムを利用するための初期設定として、他人からのタスクを受け付けない時間帯を設定する「空き時間プロファイリング」を行う。空き時間プロファイリングはCGIのプログラムとして提供されており、被験者(以下ユーザAと称す)はタスクを受け付けない時間帯(例:23:00~7:00)と対象とする曜日を最大5種類設定する。

ユーザAは認証機能を利用し、他の被験者との情報交換に関する相互認証を行う。ここでは相互認証を結んだ相手をユーザBおよびCとする。各ユーザは個人タスクをいくつか登録しているものとする。ここで、ユーザAがBおよびCとの協調タスクを登録する場合、ユーザAはクライアントの「協調ユーザ」機能より、ユーザB、Cを指定する。クライアントはサーバと通信し、ユーザBおよびユーザCのタスク情報を取得する。情報が取得された状態を図4.17に示す。この際のタスク情報はアクセスコントロールがかけられており、1月24日に注目すると、ユーザBについてはすでに登録されているユーザAとの協調タスク「ミーティング(10時~12時)」の情報および空き時間(12時~18時)が公開される。ユーザCについてはタスク情報は一切公開されず、空き時間(16時~20時)のみが公開される。

ここで、ユーザAは新たな協調タスクを設定可能な時間帯を選択する。ユーザAが日付のフォーカスを上下に移動させると、クライアントは3名に共通の空き時間帯にフォーカスをあてる。この場合は1月24日の16時~18時、あるいは1月29日の12時から14時が候補となる。ユーザAはどちらかの時間帯を選択し、決定ボタンを押すと、タスクの登録画面に移動

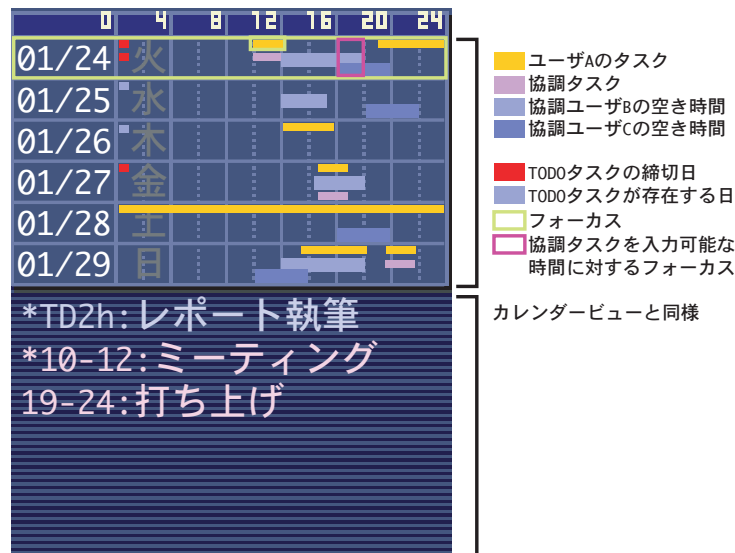


図 4.17 協調タスクスケジューリングの実例

する．この画面では協調タスクの相手および時間帯などの主要な情報があらかじめ入力された状態となっている．ユーザ A は新たな協調タスクのタスク名等を入力し，サーバに送信する．

サーバは，登録された協調タスクの情報をもとに，データベース内の各ユーザのデータを更新し，ユーザ B および C にメールを送付する．このメールにはクライアントを起動し同期を行うためのリンクが埋め込まれており，ユーザ B および C がクライアントを起動すると同期処理が実行され，新たな協調タスクが登録された旨が表示される．ユーザ B および C はそれに同意するかどうかを選択し，両者が同意した場合には正式にタスク情報が登録される．このタスク情報はユーザ B および C が改めて入力し直す必要はない．こういったタスクが多ければ，全体としてのタスクの入力作業は軽減される．なお，1 人でも非同意を選択した場合にはバックトラッキング処理が実行され，ユーザ B および C にはタスク往訪は登録されない．また，ユーザ A が次回に同期作業を行う際には非同意された旨が表示され，タスク情報が削除される．

被験者はこういったタスク登録作業を繰り返すことで，個々のスケジュールを作成していくことになる．定期的な協調タスクが存在する場合には，一括登録が可能であるため，タスクの入力回数はさらに減ることが期待される．

実験期間中に登録された個人タスクの平均は 73.3 件であった．また協調タスクは計 62 件であった．この協調タスク・依頼タスクは平均 3.0 名に関係しており，個人タスクの 28.3% を占める．また，共同作業タスクの入力支援機能により，1 名あたり 13.9 件のタスク入力の手間を

軽減している。

なお、62件の協調タスクにおいては、メンバーの拒否による再登録や交渉は1度も発生しなかった。限られた環境ではあるが、本実証実験においてはあらかじめ情報を共有することで協調的なスケジューリングを行う手法が有効であることが確認された。

4.2.2 グループ発見アルゴリズムの評価

実証実験で得られたタスクの依頼関係ネットワークを図4.18に示す。このようにタスクの依頼関係は極めて複雑なものになる。このネットワークに対し前述のグループ発見アルゴリズムを適用すると図4.18(a)の楕円に示すような複数のグループが得られる。また、特定のユーザを含むグループの一覧を図4.18(b)に示す。このように、提案手法によってあるユーザが複数のグループに所属していることがわかる。ここで、提案手法が発見したグループが適切であるかどうかを確認するため、被験者のそれぞれについて本人を含むグループのリストを提示し、評価を行った。この実験では、個々のグループに名前をつけてもらい、正しくつけられたものを適切であるとし、リストに含まれる名前が多すぎる、もしくは少なすぎると指摘されたものについては発見されたグループが適切でないとした。

被験者に提示したグループはのべ24個(計9グループ)である。うち有効回答は21個(8名分)である。結果は20個(95.2%)のグループについて適切であるとの回答を得た。適切でないとされたグループについては構成員の数が1名足りないと指摘された。また、被験者の1人は、提示された5名を含むグループのうちの3名で構成されるもうひとつのグループが存在すると回答した。このような包含関係にあるグループの発見は現状のアルゴリズムでは解決できない。タスク依頼の頻度によるクラスタリングといった新しい手法が必要になると考えられる。

提案システムによって発見された9グループの中には、被験者のメンバー選定の段階では考慮していなかったグループ(3名)が含まれる。このグループに所属する全員が適切であると回答していることから、本システムのグループ発見アルゴリズムは有効に機能しているといえる。

4.2.3 情報フィルタリングの評価

次に、情報フィルタリングの効果を検討するため、全ての被験者に対してそれぞれが入力したタスクの一覧およびそれらのタスクがどのユーザに閲覧可能な状態であったかをリスト化したものを提示した。その際には結果に対して同意または非同意のチェックを入れてもらい、備

表 4.1 フィルタリング手法の評価

	同意	非同意
Evaluation	484 (80.1%)	120 (19.9%)
Remark 1	0	96 (15.9%)
Remark 2	0	11 (1.8%)
Remark 3	159	13 (2.2%)
Remark 4	30	0
Remark 5	48	0
None	247	0

考として「1. 誰にも見せたくなかった」「2. 一部のユーザだけに見せたかった」「3. 全員が見てもよかった」「4. もっと見せたいユーザがいた」「5. 全員に見せたかった」という5種類の選択肢を用意した。各タスクに対するチェックは必須とし、備考は前述の質問で「非同意」であったものは必須、「同意」であるものは任意で付加してもらうものとした。

被験者に提示したタスクの総数は660件である。うち604件(8名分)について回答を得た。結果を表4.1に示す。必須項目の「結果に対する同意・非同意」については「同意」が484件(80.1%)、「非同意」が120件(19.9%)であった。「非同意」のうち、その理由として備考の「1. 誰にも見せたくなかった」を挙げたものは96件である。これは有効回答全体の15.9%、「非同意」の80.0%を占める。本システムでは、4.1節に述べたように各タスクに対して情報の公開レベルを「全てのユーザに公開しない」に設定することができる。そのため、これらのタスクについてはシステムとして問題の解決が容易である。同様に、非同意の理由が「3. 全員が見てもよかった」であった13件(全体の2.2%、「非同意」の10.8%)についても情報公開レベルを「全てのユーザに公開する」とすることで解決できる。しかしながら、非同意の理由が「2. 一部のユーザだけに見せたかった」すなわち閲覧可能なユーザが多すぎるとされる11件(全体の1.8%、「非同意」の9.2%)については、前節のグループ発見アルゴリズムに関する問題点と同様に別の手法によるフィルタリングや手動設定が必要になると思われる。これらの解消は今後の課題である。

全体としては、「同意」もしくは解決が容易な「非同意」タスクは593件、有効回答全体の98.2%となっており、提案手法によるスケジューリング支援は有効に機能しているといえる。

表 4.2 フィルタリング手法の性能

タスクの関係者数	タスク数	提案手法による拡張
2	50	49
3	33	14
4	35	30
5	45	0
合計	163	93

4.2.4 情報フィルタリングの性能

グループ発見の結果をもとに情報のフィルタリングを行う提案手法が、どの程度のアクセスコントロールを実現したかを検証するために、前節の評価結果の分析を行った。

検証方法は、提案手法を用いずに、各協調タスクの関係者の間でのみ情報共有を行う場合と、提案手法に基づきグループ内での情報共有を行う場合の両方について、アクセス可能なメンバーが拡張されたかどうかを比較する。ここでの協調タスクは前節において「適切な情報公開がなされた」と回答されたもののみを選んでいくことから^{*1}、グループ内の情報共有によってメンバーが拡張された場合には、フィルタリングの条件が適切に緩和されたとみなすことができる。この比較にあたっては、協調タスクの関係者が何人であったかによって分類を行った。結果を表 4.2 に示す。

協調タスクの関係者が 2 名である場合には、ほぼ確実に情報公開の範囲が拡張された。先に述べたように、これらのタスクの情報公開は適切であるとの回答を得ているため、提案手法による情報共有は有効であるといえる。関係者が 3 名の場合には、情報共有の範囲が拡大されるためにはより大きなグループが同定される必要があるため、2 名の場合と比較して減少傾向が見られる。一般的には、協調タスクの関係者が増加するにしたがってこの減少傾向が続くと思われるが、今回の実証実験では 4 名の関係者を持つ協調タスクの情報が、5 名のグループに対して公開される例が頻発したために単調減少にはならない。関係者が 5 名である協調タスクについては、実証実験において 6 名以上から形成されるグループが発見できなかったために、情報公開の範囲の拡張は起こらなかった。

^{*1} 本実験では、すべての協調タスクについて「情報公開が適切ではなかった」と回答されたものは存在しなかった。

このように、極大完全部分グラフの抽出によるグループ発見手法においては、所属するユーザの多いグループを正確に抽出することが難しい。そのため、システム全体のユーザ数が増加するにつれ、前節の評価において「非同意」であった理由が「4. もっと見せたいユーザがいた」となるようなタスクが増加すると予想される。このような問題を解決するためには、グループ発見の条件を緩和する必要がある。ただし、条件の緩和はユーザの予期しない情報公開につながる恐れがあるため、これらのトレードオフを考慮した手法を導入する必要がある。これについては今後の課題とする。

4.2.5 アンケート調査

実証実験の終了後、被験者がこれまでに利用していたスケジューリング支援ツールと本システムの性質を比較し、本システムが個人の活動を支援しているかを確認するため、被験者に対してアンケート調査を行った。アンケートの回答者数は被験者の全員である16名である。質問内容と主な回答を以下に述べる。

- 質問：タスクの入力数は増加したか？減少したか？

12名が従来のツールに入力していたタスクの数とほぼ同数か、それよりも多いと回答した。とくに、関係が密なグループでは協調タスクの数が多く、そのために入力し忘れていたタスクが他のユーザによって入力されているといった場合が多いとの回答を得た。また、PDAや手帳と比較して携帯端末を持ち歩いている時間が長いため、規模の小さいタスクを入力することが増えたとの回答もあった。

- 質問：提案システムの特徴はどこにあると思うか？

被験者には本システムのマニュアルを配布していたため、機能のあらましについては理解していたが、その中でも印象に残った機能をあげてもらった。とくに回答が多かったのはスケジュールデータの共有と閲覧、協調タスクの投入機能についてのものであった。上でも述べたように、入力を忘れていたタスクが他のユーザにとって補完されるという部分がこれまでにない機能であるとして好評を得た。また、情報量の多いインターフェイス、とくにカレンダービューについても好意的な意見が寄せられた。

- 質問：提案システムに足りない機能は何か？

この質問では携帯電話にデフォルト設定として入っているカレンダーと比較した意見があると予想していたが、実際にこれらの機能を活用しているユーザがほとんどいなかったため、回答としては協調機能の一層の充実を求める意見が大半を占めた。具体的には、協調・依頼タスクにおけるタスク実行者リストの一覧性を高めることや、依頼関係

によって構築されたグループがどのようなものであるかということが見えるような仕組みが求められていた。

- 質問：今後も提案システムを使い続けるか？

11名の被験者がこの質問に Yes と答えた。ただし、多くの友人が使うようになれば、という意見が多く、本システムがコミュニティ活動を支援するものとして利用されていることが明確になった。

4.3 考察

本研究によって明らかになった知見を以下に述べる。

4.3.1 対象となるコミュニティ

本システムが対象としているのは会社組織のようにあらかじめ上下関係が定められたグループではない。このようなグループにおいては、上位の階層に属する人間が下位の階層の人間のスケジュールをある程度自由にコントロールすることが可能であるために、意思決定までのトータルコストが非常に低い。しかしながら、階層が明確でないようなグループでは、各構成員の上下関係がないために、一旦リソースの競合が発生するとそれを解決するための交渉や妥協が必要になる。この競合解消に非常にコストがかかるために、このようなグループ形態は営利企業等では採用されないことが多い。

本システムでは、グループの構成員全てがあらかじめ自身のリソース情報を公開することを前提としている。このような環境では、他人の時間リソースの状況を考慮した上でリソースの予約をすることが可能になるために、予期しないリソースの競合を引き起こすことが少なくなる。また、競合が実際に発生したとしても、交渉のサイクルが短くなると思われる。このモデルによって競合が全く発生しなくなることはないが、フラットな組織を維持するためのコストを大きく低減できると考えられる。

本システムでは、メンバーリストの作成や管理者の決定といったグループの定義をあらかじめ行う必要はない。ユーザは、他のユーザに対して協調・依頼タスクを発行するという操作のみを行い、そこからの関係の発見やグループの同定はすべてサーバ側が自動的に行う。これにより、ボトムアップに構築されるグループや、タスクの発生とともに生まれるアドホックなグループに対しても適切な支援が可能になる。

このように、本研究が対象としているユーザ層は、既存のグループウェアモデルによる管理が難しい組織に属する人々である。しかしながら、多くの人々は、複数所属する組織の中で、

一部は階層型，その他はフラット型といったように，組織の構成が複合化している場合が多い．その際には，すべてのリソース管理を提案システムのみで行うことは適切でない．グループウェアモデルと提案モデルの連携が必要であると思われるが，これは今後の課題である．

4.3.2 情報フィルタリングの効用

本システムは，各ユーザの自発的な情報公開がなされることによってはじめて機能する．システムの有用性は情報公開へのインセンティブになりうるが，プライバシーの侵害や過剰な情報公開への危惧が大きな阻害要因となる．そこで，本システムではタスク情報の自動的なフィルタリング機能を実装し，望まない情報公開や全ての認証ユーザに全ての情報を公開してしまうといった状況を回避することが可能である．実際には，タスクの依頼関係からグループを同定し，グループをまたぐような情報アクセスを禁ずる．これにより，プライバシーの侵害を最小限に抑えた上で，情報共有によるメリットを享受することが可能になる．

しかしながら，提案システムにおいては同定されたグループの情報についてユーザに提示しなかったことから，システムがユーザの状況をどのように判定し，フィルタリングを行ったかが理解できないといった不安の声が聞かれた．このような問題を解決するために，ユーザインターフェイスを改良し，ユーザのリクエストに応じて適切な情報を可視化する必要があると思われる．

また，提案手法にはパーソナルネットワークから極大完全グラフとなる部分グラフを発見するアルゴリズムが含まれているが，これは予備実験からの経験則によって採用したものである．本手法では大規模なグループを抽出することが困難であることから，巨大な社会ネットワーク上において，より精度の高いグループ同定を行うためには，ネットワーク形成の時系列性や密度を利用し，不完全なグラフをグループとして扱う手法が必要になるとと思われる．これについては今後の課題である．

4.3.3 インターフェイス性能の向上

本システムでは，1人のクライアントソフトウェアから他の複数のユーザの持つタスクデータに直接タスク情報を書き込むことが可能である．他ユーザはこのタスク情報を自分で入力する必要がないため，トータルで見るとデータ入力のコストは大幅に低減する．携帯電話のアプリケーションを導入する上で阻害要因となるのはデータ入力の手間であることが多い．このような協調モデルには携帯電話用アプリケーションのデメリットを打ち消す効果があると考えられる．

また、従来の携帯電話用スケジューラは既存の PIM 用のインターフェイスを踏襲しているものが多い。PC や PDA のような画像解像度の高いデバイスでは一度に多くの情報を表示することが可能であるため、このようなインターフェイスでも問題が生じにくい。しかしながら、携帯電話の小さなディスプレイ上ではカレンダーを表示するだけで画面が埋められてしまい、タスクの存在が認識しにくい。本システムのクライアントソフトウェアではカレンダーの日付ごとに時間軸を付加してタスクの存在を把握しやすいようにするなど、小さなディスプレイ上に多様な情報を載せることが可能である。また、協調タスクビューによって同じタスク情報を日付ごとのガントチャート（クロノグラム）として表示し直すことや、ユーザマネージャでの認証ユーザごとのタスク分類表示など、多様な情報の見せ方を提案している。ユーザはこれらの表示をもとに自らのタスク情報について新たな知見を得ることが可能になると思われる。

ただし、前にも述べたように、フラットな組織内でのリソースの競合解消のためには交渉のステップが避けられない。本システムにおいてそのサイクルを短くすることが可能になったが、さらにその交渉過程をシンプルにする必要がある。そのためには、インターフェイスの改善や交渉モデルの簡素化が重要となる。

4.4 本章のまとめ

本研究では個人のリソース管理の手法として協調モデルを提案し、携帯電話用タスクスケジューラとしてこのモデルを実装した。

本研究が対象としているのは会社組織のようにあらかじめ上下関係が定められておらず、その範囲が明確でないようなグループに複数所属しているような人々である。

提案システムでは、既存のグループウェア等とは異なり、メンバーリストの作成や管理者の決定といったグループの定義をあらかじめ行う必要はない。ユーザは、他のユーザに対して協調・依頼タスクを発行するという操作のみを行い、そこからの関係の発見やグループの同定はすべてサーバ側が自動的に行う。これにより、ボトムアップに構築されるグループや、タスクの発生とともに生まれるアドホックなグループに対しても適切な支援が可能になる。

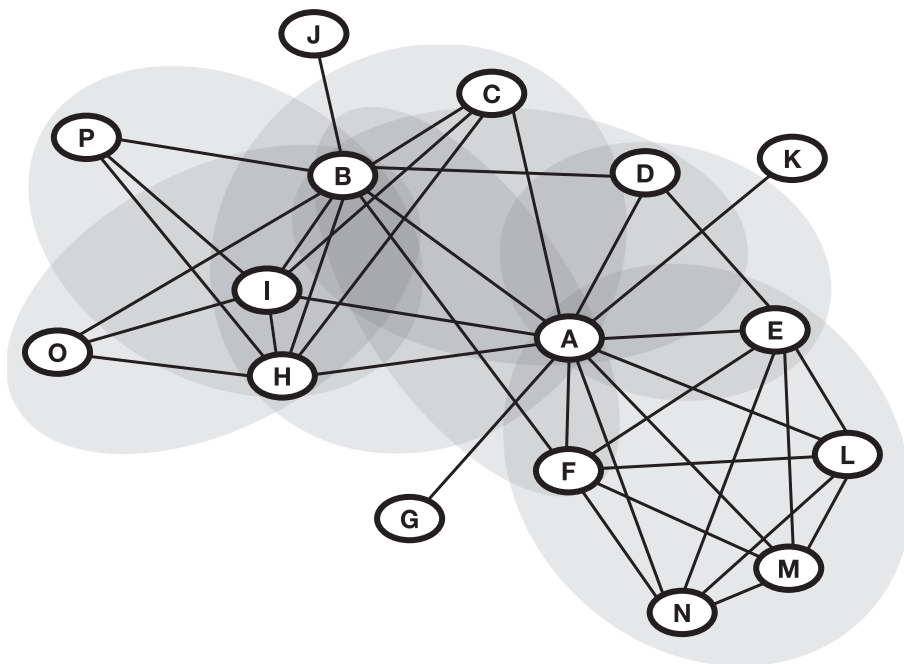
提案システムでは、各ユーザがあらかじめ自身のリソース情報を公開することを前提としている。このような環境では、他人の時間リソースの状況を考慮した上でリソースの予約をすることが可能になるために、予期しないリソースの競合を引き起こすことが少なくなる。また、競合が実際に発生したとしても、交渉のサイクルが短くなると思われる。

過剰な情報公開によるプライバシーの侵害に対しては、タスク情報のアクセスコントロール機能によって望まない情報公開や全ての認証ユーザに全ての情報を公開してしまうといった状況を回避することが可能である。実際には、タスクの依頼関係からグループを同定し、グルー

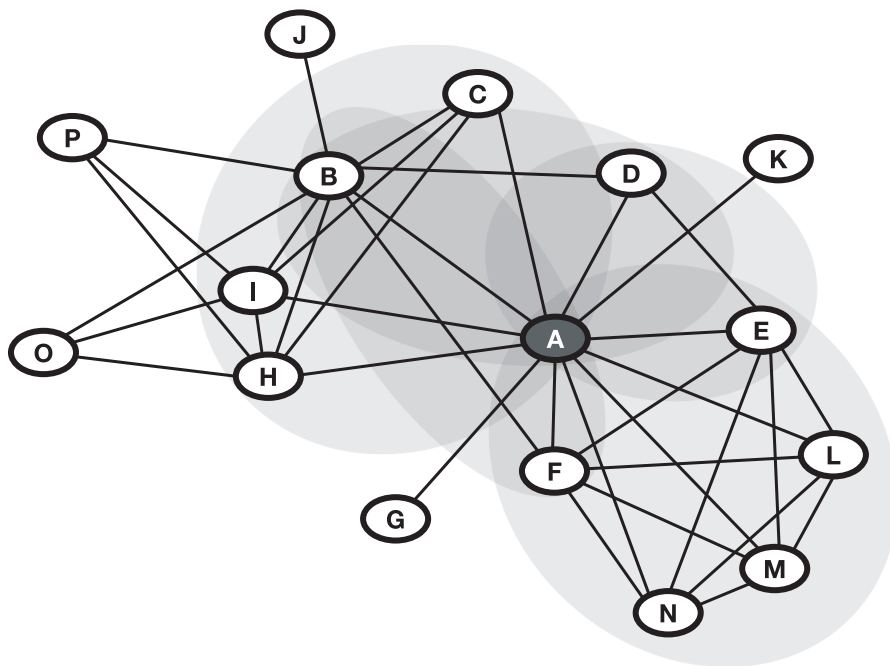
ブを横断するような情報アクセスを禁ずる。これにより、プライバシーの侵害を最小限に抑えた上で、情報共有によるメリットを享受することが可能になる。

提案システムでは、1人のクライアントソフトウェアから他の複数のユーザの持つタスクデータに直接タスク情報を書き込むことが可能である。他ユーザはこのタスク情報を自分で入力する必要がないため、トータルで見るとデータ入力のコストは大幅に低減する。携帯電話のアプリケーションを導入する上で阻害要因となるのはデータ入力の手間であることが多い。このような協調モデルには携帯電話用アプリケーションのデメリットを打ち消す効果があると考えられる。

現状のシステムでは、ユーザ間に張られたパスを消去しないために、発見されるグループの数が単調増加する。今後の課題として、タスクの依頼関係パスに時間の要素を導入し、一定期間タスクのやりとりのないユーザ間の関係を消去することや、パスの時間的要素を考慮してアクティブなグループのみを発見する手法 [23] を導入することを考える。



(a) All Groups



(b) Groups around User A



図 4.18 人間関係ネットワーク

第 5 章

Weblog とセマンティック Web による 個人の情報流通モデル

本章では、Community Web モデルの実現において重要な課題となるスケーラビリティの確保という問題について、既存の Web 環境に適用可能な手法を提案する。戦略としては、Web 上での個人性の表出のために、個人用コンテンツマネジメントシステムとして注目を集めている Weblog を利用し、個人間の情報流通を円滑にするためにセマンティック Web 技術、とくにメタデータを利用する。

本章では、要素技術を概説するとともに Community Web モデルとの関連性について述べ、それらの要素技術をいかに組み合わせるかといった基本的な方針について述べる。

5.1 研究背景

本研究では、インターネット上における個人間のコミュニケーションに焦点を当て、これを支援するプラットフォームの提案を行う。1 章で述べたように、個人間のコミュニケーションにはさまざまな内容や形態が含まれるが、本研究では個人の持つ知識の発信、流通を対象とする。とくに、近年のインターネットユーザに多く見られる、コンテンツの享受者であると同時に書き手あるいは編集者でもあるような人々に対して有効な支援を行うことを目指す。

Community Web 活動モデルに基づいてコンテンツの流通を考えると、情報の Create プロセスはそれ自体が単独で存在するのではなく、前段階における情報収集および後段階としての公開・配信プロセスを含めたサイクルとして捉える必要がある。

しかしながら、現在の Web 環境が提供するものは作成されたコンテンツを公開する場の提供、すなわち Donate プロセスの支援のみである。Collect プロセスの支援は Web ブラウザが、また Create プロセスは HTML エディタ等のアプリケーションが担っているが、これらを利用してサイクルとして組み上げるのはユーザ自身であり、システムとして統一的にサポートされているわけではない。また、Community Web 活動モデルの第 2 層であるコミュニケーション活動については支援がなされておらず、ユーザが Web を利用するにあたって暗黙的にコミュニケーションが行われているに過ぎない。

このように、現状では Web における流通プロセス全体を統合的に支援する枠組みは存在しない。そこで、本研究ではこれを実現するために、Weblog およびセマンティック Web 技術を利用し、個人のためのコンテンツ流通支援システムを設計・実装する。

5.2 Weblog とセマンティック Web

2 章で述べたように、効果的な情報流通を実現するにあたってはセマンティック Web の概念は非常に重要であるが、現時点ではこれを実現するために必要な技術的要件を満たすことが極めて難しい。また、一般のユーザがコンテンツを記述するにあたっての負荷について考慮されていないために、十分な量のコンテンツが準備されない可能性も高い。

そこで、本研究では、セマンティック Web と同じ要素技術を用いながら、Semantic Web Layer Cake とは異なるアプローチからセマンティック Web の理想である「Trust なコンテンツの流通」を実現することを目指す。

具体的には、前述の Weblog とセマンティック Web における低レベルのメタデータセットを適切に組み合わせることで、情報流通を効果的に支援し、結果として必要な情報がユーザに

到達するようなシステムを提案する。

本研究は、Berners-Lee が提唱するセマンティック Web の戦略と比較して、コンテンツの Trust に対する考え方およびセマンティックアノテーションの手法の 2 点について大きく異なる。

セマンティック Web では、コンテンツの Trust すなわち信頼性はそのコンテンツが持つ論理的な一貫性によって計算されるという立場を取っている。しかしながら、実世界ではすべてのユーザが常に論理的な一貫性のあるコンテンツを作成するとは限らない。ここではユーザのミスといった単純なものから、悪意あるユーザが意図的に誤った情報を作成するなど、さまざまなレベルの問題が起こりうる。このようなコンテンツが含まれる環境において、コンピュータによる推論機構を実行したとしても、得られる結果が正しいかどうかを判定することは困難である。また、同じコンテンツであっても、それを求めるユーザによって Trust の度合いが異なることが予想されるが、セマンティック Web のモデルでは Trust が客観的に一意に決定され、現実性を欠く。

一方、本研究においては、コンテンツの Trust とは、それがどのようなユーザによって作成されたのかという情報から得られる定性的なものであり、それぞれのユーザの Trust は、現実の生活の中で認識されるものであると考える。このモデルでは、あるユーザについて、他のユーザがそれぞれ異なる度合いの Trust を付与することは自然である。しかしながら、このような Trust はコンピュータに扱えるような形式に変換することは難しい。そこで、本研究では Trust を明示的な形で扱うのではなく、ユーザが暗黙的に認識しているものとして、他のユーザの活動を可視化することを主眼に置いた支援を行うことを考える。

提案システムを実現する上で、Weblog の存在は極めて重要である。先に述べたように、多くの Weblog は個人単位で設置され、その中でコンテンツが管理されている。このような環境では、あるサイトに含まれるコンテンツの集合が、すべて特定のユーザによって所有されていることが明確になる。これにより、Web のコンテンツをその作者と結びつけることが容易になり、その作者に対する Trust の付与をコンテンツの評価につなげるための基盤となる。

セマンティックアノテーションに関する本研究のアプローチとしては、セマンティック Web の実現プロセスのうち、第 1 段階の XML によるデータ構造の統一に関する部分を取り入れる。とくに、Semantic Web Layer Cake の低層にあたる RDF メタデータを積極的に利用する。アノテーションの手法については、自然言語処理による自動化は考えず、次に述べる Weblog ツールなどのコンテンツマネジメントシステム (CMS) を利用することで、ユーザに負担をかけることのない、確実なアノテーションを提供する。また、過去のコンテンツの引用・再利用をしやすくすることによって、こういったコンテンツについても間接的にアノテーションすることを可能にする。

本研究では、既存の要素技術に対して以下のような拡張を行う。

- **メタデータの拡張**
現在普及が進んでいるメタデータセットに対し、Weblog との連携に必要なフレームワークの提案と語彙の設計を行う。
- **メタデータマネジメントシステムの拡張**
既存のメタデータセット、および新たに提案する拡張メタデータを適切に管理するためのモデルを提案し、アプリケーションとして実装する。

また、本研究ではメタデータを活用するためのアプリケーションの提案と実装を行う。

- **Community Web 活動モデルに基づくコンテンツ収集・配信アプリケーション**
前述のメタデータを利用し、Community Web 活動モデルのすべてのプロセスを支援するためのアプリケーションを提案し、実装を行う。
- **Web アプリケーションの拡張**
拡張されたメタデータを用いてユーザに新たな利便性を提供するアプリケーションを提案し、実装を行う。

5.3 Weblog のアーキテクチャ

本研究では、現状で利用可能なセマンティック Web の諸技術および Weblog に関する技術を、4 層からなるアーキテクチャの各層に対応づける。アーキテクチャの概要を以下に示す。

- **メタデータ層**
XML および RDF を用いたメタデータフォーマットによって定型化されたコンテンツを示す。
- **マネジメント層**
メタデータおよびコンテンツを効率的に生成、管理するためのツールを示す。
- **アグリゲーション層**
他のサイトで配信されるメタデータを集約し、ユーザに提示するためのツールを示す。
- **アプリケーション層**
下層によって形成されるプラットフォーム上で新たなサービスを提供するツールを示す。

このモデルにおいて、各層に対応する利用可能な要素技術を次節以降で詳説する。

5.4 メタデータ層

5.4.1 RSS

Weblog サイトの増加にしたがって普及が進んだメタデータフォーマットの代表的が RSS である [64]。RSS は Web サイトの概要を XML 形式で記述するためのメタデータフォーマットである。プレーンな RSS は表現力に乏しいが、モジュールという形式を用いて他のメタデータフォーマットの語彙を利用することで、サイト内の各コンテンツのタイトル、更新時間などの追加情報が記述できる。

RSS は現在 0.91, 1.0, 2.0 という 3 つのバージョンが流通しており、それぞれを支持する勢力の間で議論が続いている。RSS が何の略称であるかはバージョンによって異なる。RSS 0.91 では Rich Site Summary, RSS 1.0 では RDF Site Summary, RSS 2.0 では Really Simple Syndication である。また、Weblog に適したサイト概要フォーマットとして新たに Atom^{*1} という規格が提案されている。

日本では RSS 1.0^{??}が比較的によく利用されている。以下では 1.0 を取り上げて説明する。RSS 1.0 は RDF (Resource Description Framework) [65] に基づいており、それを XML 形式で記述したものとして表現される。RSS 1.0 での記述例を図 5.1 に示す。RSS 1.0 は RDF の宣言、channel、item の 3 つのブロックからなる。RDF の宣言部分では、この XML で使用する語彙の体系であるモジュールの名前空間の URI を指定する。この例では RDF 自体の語彙、RSS 1.0 で規定された語彙とともに、書誌情報のメタデータを定義している Dublin Core Metadata Initiative^{*2}の語彙を利用する。channel にはサイト全体の概要が記述され、item にはそのサイトに含まれる個々のコンテンツあるいはエントリーに関する情報が記述される。channel および item に共通する要素としては RSS 1.0 が持っている title や link (もとのコンテンツの URI、item の場合はエントリー毎に生成されたページの URI)、description (サイトの概要。通常は HTML タグを除いた本文の先頭の数十バイト) と、Dublin Core によって定められた dc:date を用いた更新時間情報や、dc:subject を用いたユーザ指定のカテゴリ情報などがある。また、channel すなわちサイト全体のメタデータとしては、エントリーがどれだけ含まれているのかを示す items を記述する必要がある。

すでに一部の Web サイトでは RSS の配信が行われている。また、後述の Weblog ツールは HTML と同様に RSS を自動生成することが可能である。作成者等の属性はあらかじめユーザ

*1 <http://www.atomenabled.org/>

*2 <http://dublincore.org/>

に初期設定として Weblog ツールに入力させたものを埋め込み、各コンテンツの概要、更新時間および RSS が指す HTML ファイルの URI 等はコンテンツが入力された際に自動的に記述される。Weblog ツールによって、新たなコストをかけることなく RSS を配信することができるため、一般のユーザにおいてもメタデータの効用が得やすくなっている。

これを利用して各サイトが配信する RSS を収集し、これを整形することで多くの情報を短時間に閲覧するアグリゲータと呼ばれるアプリケーションやサービスも生まれている。

5.4.2 FOAF

Weblog ツールや RSS の普及と前後して、FOAF[66] というメタデータフォーマットも利用されるようになってきた。FOAF は Friend Of A Friend の略で、友人関係を表現するためのメタデータである。RSS と同様に、FOAF も 2 つのパートから構成されており、最初のパートには本人の名前やメールアドレスなどが記載されており、後半のパートには友人の名前、ID などを knows タグで記述する。FOAF ファイルは各人で分散的に管理される。これらの FOAF データを横断的に検索し、友人関係ネットワークを可視化する FOAFnaut^{*3}や、情報を集約し、ポータルサイトとして機能する PLINK^{*4}などのサービスが提供されている。現状ではそれ以上の利用法が提案されていないが、今後は後述のソーシャルネットワークとの関連から、このフォーマットが一般に普及する可能性は高い。

5.5 マネジメント層

5.5.1 Weblog ツール

現在の Weblog の普及において Weblog ツールはなくてはならない存在である。これらのツールが書き手の時間的・心理的コストを下げ、コミュニケーションの形態を変え、結果として先に述べたような数の Weblog サイトを生み出したといえる。

Weblog ツールはこれまでに述べた Weblog サイトを容易に構築することのできるソフトウェアである。Weblog サイトが登場し始めた 1998 年前後から、CGI 等を用いてブラウザからコンテンツを記述・公開するためのツールはいくつか存在した。こういったツールは徐々に高機能化し、2000 年から 2001 年にかけて BLOGGER^{*5}や Movable Type^{*6}といった代表的

*3 <http://jibbering.com/foaf/>

*4 <http://beta.plink.org/>

*5 <http://www.blogger.com/>

*6 <http://www.movabletype.org/>

```
<?xml version="1.0" encoding="UTF-8"?>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns="http://purl.org/rss/1.0/">
  Namespaces

  <channel rdf:about="http://www.semblog.org/i2k/index.rdf">
    <title>semblog-dev</title>
    <link>http://www.semblog.org/i2k/</link>
    <description></description>
    <dc:language>ja</dc:language>
    <dc:date>2004-03-17T00:16:15+09:00</dc:date>

    <items>
      <rdf:Seq>
        <rdf:li rdf:resource="http://www.semblog.org/i2k/archives/000292.html" />
        <rdf:li rdf:resource="http://www.semblog.org/i2k/archives/000277.html" />
        <rdf:li rdf:resource="http://www.semblog.org/i2k/archives/000260.html" />
        <rdf:li rdf:resource="http://www.semblog.org/i2k/archives/000251.html" />
      </rdf:Seq>
    </items>
  </channel>
  Channel

  <item rdf:about="http://www.semblog.org/i2k/archives/000292.html">
    <title>FOAF共生プラットフォーム（試案）</title>
    <link>http://www.semblog.org/i2k/archives/000292.html</link>
    <description>先日の予告から少しばかり改題して，FOAFの話をしてみる．RNA1.8に
    ようやくFOAFのマネジメント機能がついた．つけたばかりということで，メタデ
    ータとして甘いところが多々あるのは承知していて，これは議論しながら改善して
    いく所存．というかこっそりつけたFOAF機能が予想以上にメタデータコミュニティ
    の人々に注目されていることに驚いています．...</description>
    <dc:subject>semblog</dc:subject>
    <dc:creator>i2k</dc:creator>
    <dc:date>2004-03-17T00:16:15+09:00</dc:date>
    </item>
    Item

</rdf:RDF>
```

図 5.1 RSS 1.0 の記述例

な Weblog ツールがリリースされるに至り，このようなシステムを利用したサイト管理が一般化した．日本でも，以前より Web 日記のためのツールが多数あり，現在でも多くのユーザーが利用している．一方で，海外産の Weblog ツールを導入する際には文字コード等の問題があったが，有志によって日本語化パッチが提供され，初期段階での普及の一助を担った．Weblog ツールの多くは，比較的小規模な，かつ個人を対象としたコンテンツマネジメントシステム

(CMS) である。Weblog ツールは以下の 3 つのタイプに大別することができる。個々のツールの詳細は文献 [67] などにまとめられている。

- スクリプト型

ユーザが自分でサーバに設置する CGI スクリプトや PHP スクリプトである。カスタマイズの自由度が高い。ツールによっては MySQL 等の DBMS が利用できる環境でなければ動作しないものもある。例：Movable Type, Bloxom^{*7}, Nucleus^{*8}など。

- サービス型

アカウントを登録するだけで利用できる Weblog サービス。非常に多くのサービスがあり、それぞれ特徴が異なる。例：ココログ^{*9}, livedoor Blog^{*10}, はてなダイアリー^{*11}など。

- クライアント型

Windows や MacOS X 用のアプリケーションとして提供されるもの。クライアント側でコンテンツ管理を行い、HTML の生成や FTP・WebDAV によるアップロードを自動的に行う。例：Radio UserLand^{*12}, iWeblog^{*13}など。

Weblog ツールの動作を、代表的なスクリプト型ツールである Movable Type を例に取り説明する。図 2.1 は Movable Type を用いて作成した Weblog サイトの例である。

ユーザはインターネットブラウザに表示されるフォームにテキストを記述し、保存する（図 5.2 参照）。

Weblog ツールは Web ブラウザ上でのコンテンツ記述・編集を可能にし、その結果は即座に HTML 化されて公開される。多くの Weblog ツールでは、書き手は一度 View テンプレートを定義しておけばその後は HTML タグ等の記述をすることなしにコンテンツを公開することができる。これにより、情報公開のためのコストは従来の HTML マークアップと FTP 等によるファイルのアップロードによる方法と比較して劇的に低減する。このコストの低減が、スモールコンテンツの生産を可能にしているといえる。

ユーザが記述したコンテンツ（エントリと呼ぶ）は Weblog ツール内のデータベースにて個別に管理されている。サイトの基本設定（サイト名など）は別に管理されており、ユーザ

*7 <http://www.bloxom.com/>

*8 <http://nucleuscms.org/>

*9 <http://www.cocolog-nifty.com/>

*10 <http://blog.livedoor.com/>

*11 <http://d.hatena.ne.jp/>

*12 <http://radio.userland.com/>

*13 <http://www.lifli.com/>

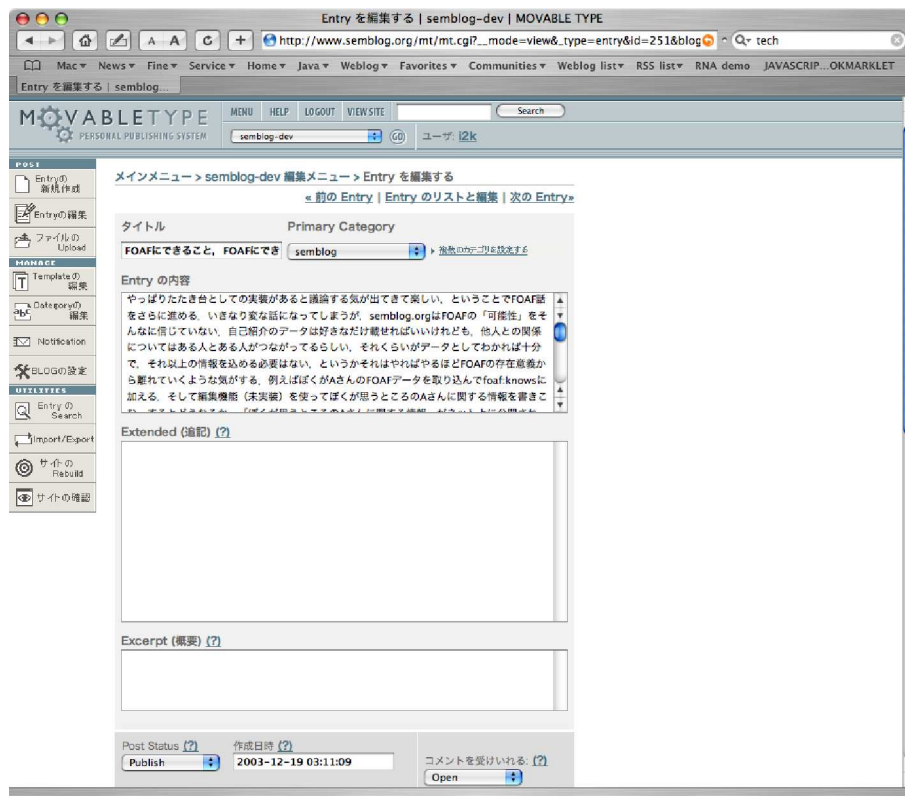


図 5.2 Movable Type におけるコンテンツ入力画面

がサイトの再構築を指示すると、Weblog ツールは各データベースより必要な情報を取得し、HTML ページを自動的に生成する（図 5.3 参照）。その際のサイトのデザインは、ひな形となるテンプレート HTML と CSS スタイルシートによって制御されており、ユーザは容易にカスタマイズすることができる。また、Weblog ツールは 1 つのエントリから単純に 1 つの HTML を生成するだけでなく、最新のエントリの複数個を並べた index ページや月ごとのアーカイブ、カテゴリごとのアーカイブなどを同時に生成する。このように、ワンソース・マルチユースを実現するシステムアーキテクチャとなっている。この特徴は、前述の RSS 配信にも生かされている。

Weblog ツールには豊富なコミュニケーション支援機能が付加されている。各エントリに対して来訪者がコメントを残すことができる掲示板機能はほぼ全ての Weblog ツールに搭載されているほか、来訪者がどのページのリンクを通じてアクセスしたかという Referrer 情報を抽出し、リンクとして表示する機能も一部のツールで用意されている。

また、Weblog ツールに特有のコミュニケーション手段として「TrackBack（トラックバック）」の存在を挙げることができる。TrackBack は Weblog サイト間の逆リンクを生成する機

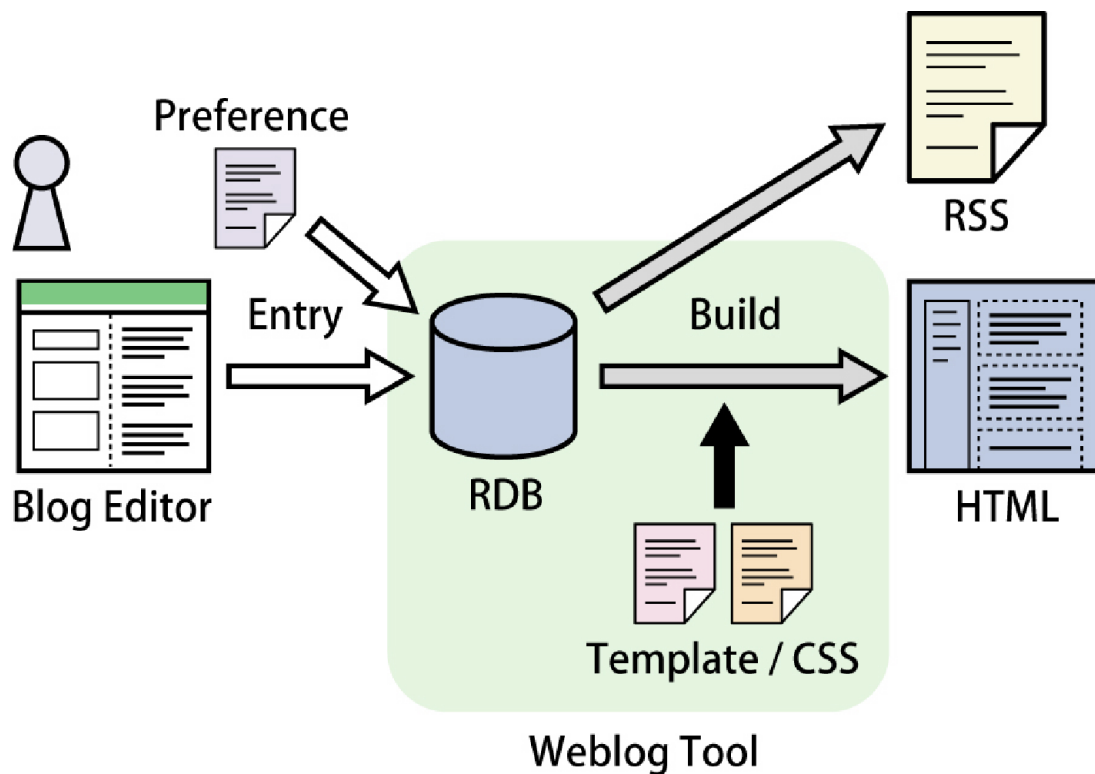


図 5.3 Weblog ツールのアーキテクチャ

構である。ユーザ A がユーザ B のサイトのコンテンツを引用してエントリを記述した際に、その旨を XML 形式の Ping メッセージとしてサイト B に通知すると、サイト B の Weblog ツールはそのメッセージを受けて B のエントリから A のエントリへのリンクを自動生成する。TrackBack によって、普通に張ることのできる順リンクとは逆方向の、相手から自分の側に来訪者を引き込むようなリンクを生成することが可能になる。この機能を利用して、引用関係を伝えることで議論の流れを時系列的に追うことが可能になる。

これまでの Web は順リンクのみで構成されているために、ページ間のリンクを移動するだけでは特定のサイトに到達できないという蝶ネクタイ構造が形成されるといわれている [68]。全てのページへ到達可能にするには、多くのページおよびリンク情報を格納したアーカイブや検索エンジンが必要となる。Weblog では TrackBack の導入によって、双方向リンクが容易に実現できる。このため複数のサイト間を行き来しながら渡り歩くようなブラウジングや、自分に関連する情報を集めるための小規模なクロールといった新しい利用法が可能になると思われる。

他にも、Weblog ツールには XML を用いたメッセージの送受信で簡易的な Web サービス

を実現する XML-RPC や、標準的な文字コードに UTF-8 を採用するなど、Web における先進的な規格や技術が盛り込まれている。これらは商用サイトにも波及し始めており、今後の Web はこういった新しい技術を下敷きに発展していくものと思われる。

5.6 アグリゲーション層

5.6.1 RSS のアグリゲーション

RSS を利用することにより，ソフトウェアを用いて複数のサイトから情報を収集し，一覧することが可能となる．これはアグリゲーションと呼ばれる．現在ではいくつかの RSS アグリゲーションソフトウェア（アグリゲータ）が提供されている．ここでは本研究で実装を行った Windows 用 RSS アグリゲータ「glucose」*¹⁴を例に取って説明する．最初に，アグリゲータのユーザは，閲覧したいサイトが配信している RSS の URI を登録する．アグリゲータは HTTP 通信によってファイルを取得する．アグリゲータによって取得された RSS は展開され，3 ペインのインターフェイスによって表示される（図 5.4 参照）．左ペインは RSS を配信するサイトのリスト（チャンネル）である．右上のペインには各コンテンツのタイトル，更新日時，サイト名等のリストが表示されており，各項目によってソートすることが可能である．右下のペインには選択されたコンテンツの内容が表示される．これらのコンテンツは一定時間ごとに自動的に再取得される．ブロードバンド環境の普及にともない，アグリゲータによる情報の閲覧はプッシュ型モデルに近づいているといえる．

日本では，Web 日記の普及と同時に，特定のサイトを登録し，更新状況をチェックするための「アンテナ」アプリケーションが利用されてきた．アンテナは各サイトに対して HTTP 通信を用いてファイルの更新時間を取得し，リストを並べ替える．動的なリンク集であるといえる．アンテナと比較すると，アグリゲータでは規格化された RSS をデータ交換フォーマットとして用いることにより，コンテンツのタイトルや作成者，概要といったよりリッチなコンテンツを取得・表示することができる．

5.7 アプリケーション層

RSS に関連するサービスとして，RSS 全文検索エンジンが存在する．海外では Feedster*¹⁵，日本では Bulkfeeds*¹⁶や FeedBack*¹⁷がその代表例である．RSS 全文検索エンジンは従来のもとの利用法は変わらないが，検索の対象が RSS のみであり，また検索結果も RSS で提供される．このサービスを利用して，RSS アグリゲータにクエリー（の URI）を登録しておく

*¹⁴ <http://www.semblog.org/glucose/>

*¹⁵ <http://www.feedster.com/>

*¹⁶ <http://bulkfeeds.net/>

*¹⁷ <http://naoya.dyndns.org/feedback/>

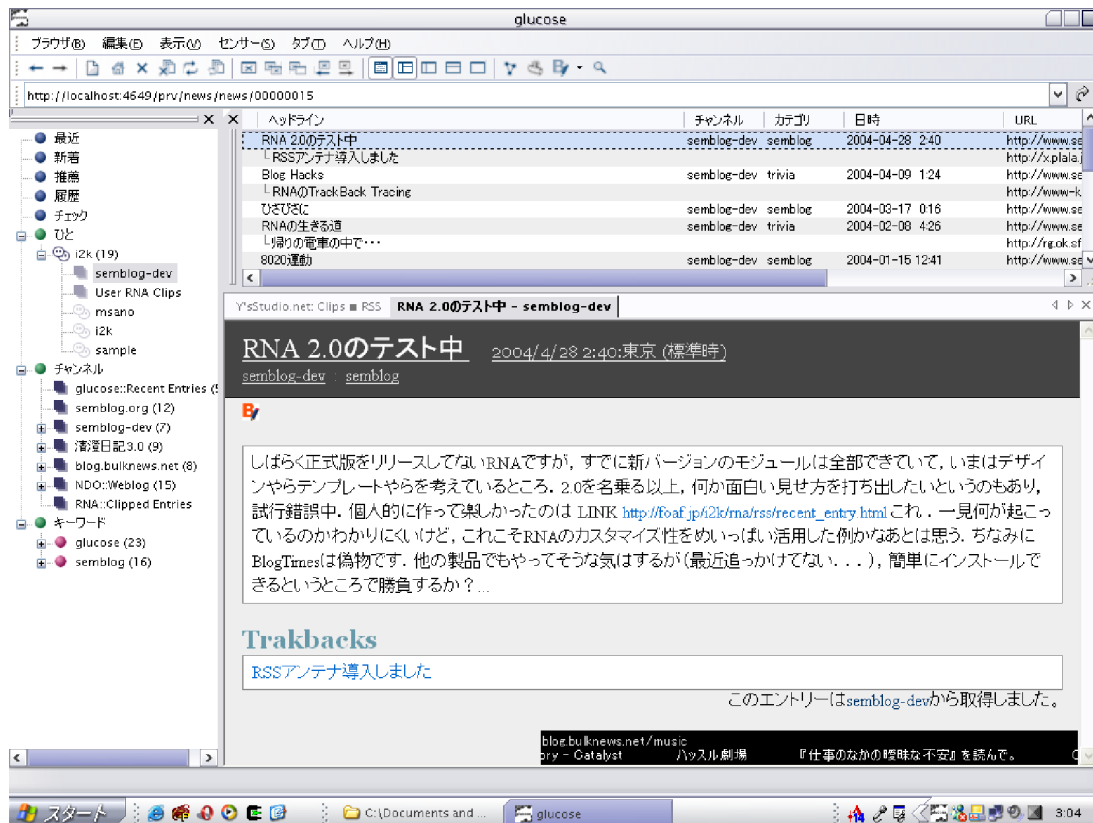


図 5.4 RSS アグリゲータ「glucose」

キーワードに合致する最新ニュースやエントリを常にチェックするなどの使用法が可能になる。

また、RSS をアーカイブするサービスの中には、引用回数の多いエントリを推薦する blogdex^{*18}や、ISBN コードを用いて本の批評を抜き出して表示する All Consuming^{*19}などがある。現状では RSS を配信しているサイト数が少ないため、コンテンツに偏りがあることは否めないが、今後さらに Weblog サイトの数が増加していけば信頼性が高まり、より多様なサービスが提供されるものと思われる。

このように、RSS によって情報の取得コストは減少したと思われるが、その際に情報の選別は行われておらず、結果として得られた情報には多くのノイズが含まれていると思われる。また、上記の枠組みには新たな情報の生産活動に結びつけるといった視点がないために、先に述べた情報流通プロセス全体を支援しているとは言えない。

*18 <http://blogdex.net/>

*19 <http://www.allconsuming.com/>

5.8 本章のまとめ

本章では、第 2 の研究対象として Web 上におけるコンテンツの流通支援環境について検討を行った。まず、近年注目されている Weblog の社会的および技術的側面について概説し、提案手法の実現可能性を示した。そのうえで、Weblog が内包する問題点を解決する可能性のあるセマンティック Web について述べ、両者の融合によって効果的な情報流通が可能になることを指摘した。また、現存の技術の中で利用可能なものを本研究の観点から分類し、それぞれの拡張の可能性について述べた。

次章では、個人のための情報流通プラットフォームを実現するために、各要素技術の拡張ならびに統合のモデルを示し、実装を行う。また、提案システムの実運用についても述べる。

第6章

個人のための情報流通プラットフォーム

本章では、実運用可能な情報流通環境を構築するために、前章で述べたメタデータ、マネジメント、アグリゲーションならびにアプリケーションの各層に属する要素技術を拡張し、それぞれを統合した Semblog プラットフォームについて述べる。

また、Semblog プラットフォームの有効性を検証するために、実際の学会コミュニティで運用した例を取り上げ、今後の展開を含めた議論を行う。

6.1 Semblog プラットフォーム

3 章および 4 章で取り上げた Community Web における情報流通の問題点，そして 5 章で述べた Web における情報配信技術や情報獲得技術の進展を下敷きに，本章では情報収集から生産，公開までを統合的に支援し，個人単位の情報流通をさらに活性化させるべく，新たな情報・コミュニケーション支援環境である「Semblog プラットフォーム」の提案を行う．

Semblog プラットフォームでは，セマンティック Web および Weblog の要素技術を利用・拡張し，Community Web 活動モデルの全体をサポートする情報システムアーキテクチャを提案する．また，プラットフォームの実現を促進するためのアプリケーションを構築し，実世界での検証を行う．

Semblog プラットフォームの構築に際しては，先に述べた Weblog の 4 層モデルに則り，各層ごとに要素技術の拡張および新規ソフトウェアの提案を行う．プラットフォームのインフラストラクチャーとなるメタデータについてはデータモデルの拡張や新しい語彙の導入を行う．また，これに伴うマネジメント用アプリケーションを提案する．ユーザとの直接のインタラクションを行うアグリゲーション層およびアプリケーション層においては，Community Web 活動モデルの実現のために新たなアプリケーションを提案，実装し，ユーザの活動がどのように支援されるかについて検討する．また，これらのアプリケーションを実際のサービスとして運用し，定性的な評価を得ることや，Semblog プラットフォームを実世界に導入するにあたっての社会的活動についても述べる．

これまでも述べたように，Semblog プラットフォームの目標は，Web において個人の存在を明確にすることである．これにより，HTML ページ単位であった Web コンテンツを個人の単位で集約し，粒度を高めることが期待されるほか，コンテンツの書き手，読み手，あるいは編集主体としての個人同士が相互にコミュニケーションを行うための基盤を提供することが可能になる．また，個人が提供するコンテンツの多様化を進めることや，コンテンツ間のリンクに対する詳細化を実現することを目指す．

以下に Semblog プラットフォームが提供する機能の概説と，その目的を挙げる．

- メタデータ層

メタデータ層では，Weblog ツールによる自動的なアノテーションに適したメタデータセットを提案する．

- PermaRSS

RSS の拡張として，Weblog ツールが管理している情報をメタデータとして定義

- し、配信する。また、現在の Weblog ツールによる RSS 配信の問題点を解消するモデルを構築する。
- FOAF
コンテンツとその書き手に関する情報とをメタデータレベルで関連づける方法として、RSS と FOAF の統合モデルを提案する。
 - マネジメント層
マネジメント層では、メタデータ層で新たに提案されたメタデータフォーマットを効率的に管理するためのアプリケーションの提案および拡張を行う。
 - Weblog ツールの拡張
既存の Weblog ツールに対し、PermaRSS および FOAF を取り扱うことができるような拡張機能を付加する。
 - FOAF TrackBack
FOAF メタデータによるパーソナルネットワークの構築を支援するために、ユーザの利便性を向上させるアプリケーションを提案する。
 - アグリケーション層
アグリゲーション層では、メタデータを利用した個人用コンテンツ収集・配信環境を実現するためのアプリケーション 2 種を設計し、実装する。
 - RNA
メタデータを利用した個人用ポータルとして機能する Web アプリケーション「RNA」を設計し、実装を行う。
 - glucose
クライアント側でメタデータを収集するとともに、Weblog ツールとの連携によって配信をサポートする Windows 用アプリケーション「glucose」の設計および実装を行う。
 - アプリケーション層
アプリケーション層では、メタデータの流通環境においてユーザに新たなサービスを提供するための検索・推薦アルゴリズムの提案と実装を行う。
 - エゴセントリック検索
Weblog の個人性、ネットワーク性を利用し、コミュニケーションを活性化させるためのローカル検索アルゴリズムの提案と実装を行う。
 - RNA アライアンス
個人が持つ複数の Web アプリケーションの連携により、パーソナルネットワーク上でのコンテンツ推薦を行う。

– パーソナルオントロジー

セマンティック Web におけるオントロジー構築をボトムアップに行うフレームワークを、メタデータやそのマネジメントの観点から提案する。

また、Semblog の応用事例として、実世界での運用例を挙げる。

- 人工知能学会全国大会 Weblog 情報支援
- RSS リーダーの実用化
- RSS に関するコミュニティ

6.1.1 Degree of Interest

本研究では、情報収集および情報発信に際して”Check”、”Clip”、および”Post”という3レベルの興味の度合い (Degree of Interest) を定義し、興味の強さに応じて情報の配信プロセスを変える。

最も弱い”Check”レベルとは、ユーザーが特定の Web サイトや情報ソースに日常的にアクセスすることを意味する。ユーザはその Web サイトのコンテンツ内容をあらかじめ知っているわけではないが、過去の更新履歴からどのような情報が掲載されうるかを知っている。本研究では、このような知識が情報流通において重要な働きを持つと考え、ユーザが日常的に巡回する Web サイトのリストを公開することで、そのユーザがどの分野に興味を持っているかを表明するための支援を行う。このリストには、サイトの URI やタイトルの他に、サイトに含まれるコンテンツの概要が記述されている。概要部分は登録先のサイトが更新するたびに变化するため、リスト自体が動的なコンテンツとして他の閲覧者によって頻繁にアクセスされる可能性が高まる。

次の”Clip”レベルとは、ユーザが閲覧したコンテンツの中でとくに興味があったものを指定し、保存することで、後日同じコンテンツに再びアクセスしやすいようにすることを意味する。本システムでは、”Check”レベルで登録された Web サイトに含まれるコンテンツの中でユーザがとくに興味があるものを指定することで、指定されたコンテンツのリストを作成、公開する支援を行う。Clip レベルの情報は個々のコンテンツへの興味を表しているため、”Check”レベルでのサイト全般へのリンクよりも強い意志を表明していると考えられる。また、Check レベルのリンクはリンク先の内容が日々変わっていくが、Clip レベルのリンクは Permalink と呼ばれる永続的なリンクであり、内容が変化しない。

同様の目的を持つものにオンラインブックマークが挙げられる。オンラインブックマークは

Web ブラウザの「ブックマーク」「お気に入り」を Web 上に作成もしくはバックアップするもので、ユーザはこれによりどの端末からでも自分のブックマークを閲覧することができる。また、他人からのアクセスの許可、不許可を選ぶことが可能であり、公開する場合にはオンラインブックマーク自体が一種の Web コンテンツとなる。しかしながら、オンラインブックマークは単純な URI のリストでしかなく、コンテンツとして捉えた場合には変化が乏しい。そのため、閲覧者にとって有益なものにはなりにくい。

最後の”Post”レベルとは、非常に強い興味を持つコンテンツに対してそれを引用し、コメントを付加して新しい情報として発信することを意味する。ここでは、単なる興味だけでなく、それに伴う意見の表明がなされる。本システムにおいては、”Post”レベルの情報発信は Weblog ツールが担うものとし、その前段階の情報収集プロセスについての支援を考える。

6.2 システム構成

Semblog プラットフォームの構成を図 6.1 に示す。本システムはサービス型とクライアント型の 2 種の RSS アグリゲータおよび検索用プログラム、そして Weblog ツールから構成される。個々のモジュールは RSS によってデータの交換が行われる。また、動的に他のモジュールを呼び出す場合には XML-RPC プロトコル [69] による通信を行う。Weblog ツールには MovableType[70] などの既存のシステムを利用する。これらのツールは RSS ならびに XML-RPC をサポートしている。

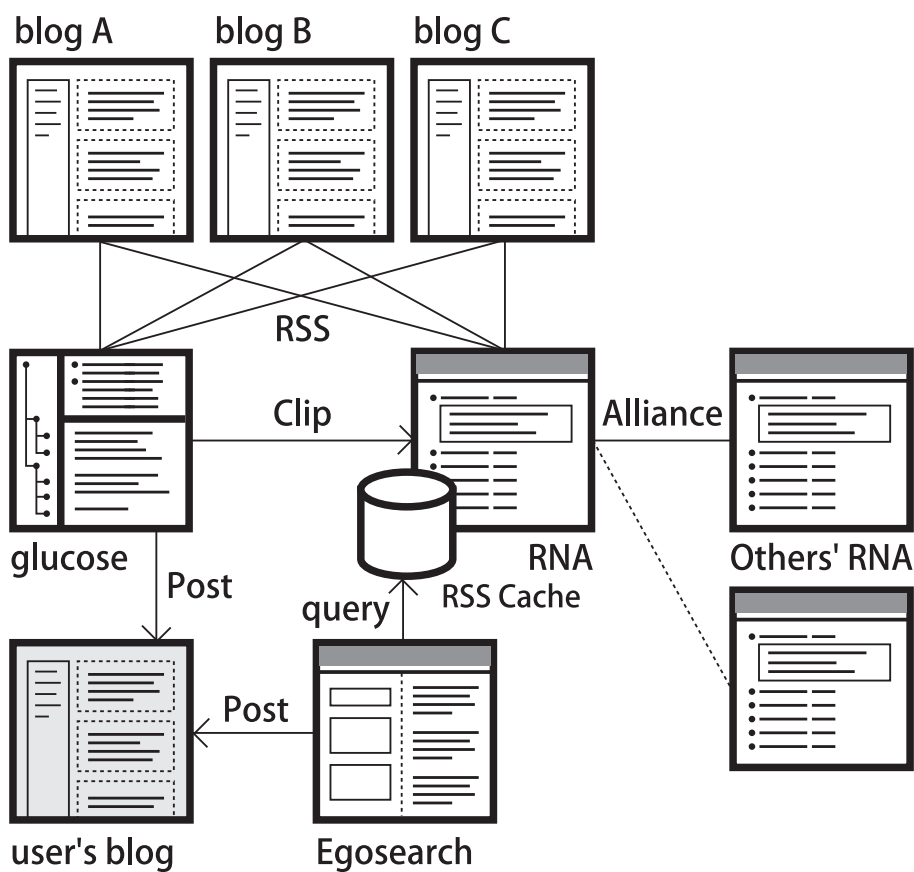


図 6.1 System Architecture

6.3 PermaRSS

Semblog プラットフォームでは、すべてのコンテンツは RSS メタデータを持つことが望ましい。しかしながら、RSS は RDF Site Summary の略であるように、サイト全体の概要、とくに最新情報を公開するためのフォーマットとして提案されたものである。RSS の規約によれば、1 つのサイトにつき 1 つの RSS を用意し、配信すべきであるとされている。これに従えば、1 つのサイトが保持するコンテンツのすべてに RSS メタデータを付加する場合には、1 つの RSS ファイルにすべてのメタデータを記述する必要がある。このような方法では、RSS ファイルのサイズが極めて大きくなり、情報流通の過程で他のアプリケーションが処理することを考慮すると現実的ではない。また、ユーザにとっては不必要な情報が流通することになり、有効な支援につながらない可能性がある。

そこで、本研究では、RSS のモデルを継承し、コンテンツ単位でのメタデータ付加を可能にするメタデータセットである「Permanent RSS: PermaRSS」を提案する。PermaRSS では、Weblog を用いたコミュニケーションに適した語彙を導入し、アグリゲーション層やアプリケーション層のツールに新たな機能を提供することを考える。

PermaRSS は、RSS とは異なり、Weblog サイトが保持する個別のコンテンツ (Permalink) に対して 1 つのファイルを対応させる。すなわち、サイトが持つコンテンツ数と PermaRSS ファイルの数は一致する。

PermaRSS のデータ構造を図 6.2 以下に示す。

PermaRSS は RSS 1.0 と同様に RDF 記法を用いて表現する。基本構造は RSS と同じく、RDF の宣言、channel、item の 3 つのブロックから構成される。RDF の宣言部では、RSS 1.0 で使用されている RDF 自体の語彙、Dublin Core に加え、Semblog プラットフォーム独自の語彙を示す名前空間である「semblog」を記述する。channel にはメタデータを付加する対象のコンテンツ情報を記述する。従来の RSS 1.0 の item 部に記述されていた title 要素、link 要素ならびに description 要素などがこの channel 要素に入る。また、PermaRSS ファイルとサイトの関係を示すために、新たな語彙 semblog:site を定義し、この要素にサイトの URI を記述する。

item 要素には、対象となるコンテンツとリンク関係を持つコンテンツに関する情報が記述される。title 要素、link 要素はそのコンテンツのタイトルと URI が入る。そして、そのコンテンツが元のコンテンツからリンクされるものであれば semblog:out 要素を、TrackBack されたものであれば semblog:in 要素を利用してコンテンツの URI を記述する。

PermaRSS の生成と管理は、既存の Weblog ツールにプラグインを付加することで行うマ

ネジメント層による方法と、RSSリーダーがRSSを収集する際にPermaRSSを生成するアプリケーション層による方法を提案する。詳細は次節以降で述べる。

```
<?xml version="1.0" encoding="UTF-8"?>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:semblog="http://www.semblog.org/ns/"
  xmlns="http://purl.org/rss/1.0/">
  Namespaces

  <channel rdf:about="http://www.semblog.org/i2k/permarss/000292.rdf">
    <title>FOAF共生プラットフォーム（試案）</title>
    <link>http://www.semblog.org/i2k/archives/000292.html</link>
    <description>先日の予告から少しばかり改題して、FOAFの話をしてみる。RNA1.8に
    ようやくFOAFのマネジメント機能がついた。つけたばかりということで、メタデ
    ータとして甘いところが多々あるのは承知していて、これは議論しながら改善して
    いく所存。というかこっそりつけたFOAF機能が予想以上にメタデータコミュニティ
    の人々に注目されていることに驚いています。...</description>
    <dc:subject>semblog</dc:subject>
    <dc:creator>i2k</dc:creator>
    <dc:date>2004-03-17T00:16:15+09:00</dc:date>

    <items>
      <rdf:Seq>
        <rdf:li rdf:resource="http://www-kasm.nii.ac.jp/archives/000084.html" />
        <rdf:li rdf:resource="http://foaf.jp/research/archives/000011.html" />
        <rdf:li rdf:resource="http://www.asahi.com/business/1107/001.html" />
      </rdf:Seq>
    </items>

    Channel
  </channel>

  <item rdf:about="http://foaf.jp/research/archives/000011.html">
    <title>ありがとうございました</title>
    <link>http://foaf.jp/research/archives/000011.html</link>
    <semblog:in>http://foaf.jp/research/archives/000011.html</semblog:in>
    <description>第3回blog勉強会は無事に終了しました。70名を超える参加者のみな
    さまに来ていただくことができ、主催者一同感謝しております。次回（いつになる
    かはわかりませんが）もよろしくお願ひします。...</description>
    <dc:creator>blog research community</dc:creator>
    <dc:date>2004-03-14T15:23:47+09:00</dc:date>

    Item
  </item>

</rdf:RDF>
```

図 6.2 PermaRSS の記述例

6.4 FOAF

Semblog プラットフォームでは、メタデータを用いて個人単位のコンテンツ管理を実現することを目指している。FOAF は個人に関する情報を定義する手段として注目が集まっているが、本研究ではこの FOAF と RSS を連携させることで上記の目標を達成することを目指す。

FOAF の規格では、個人が持つ Weblog の URI を登録するための `foaf:weblog` という語彙が用意されている。しかしながら、メタデータを扱うアプリケーションの観点では、FOAF 内の `foaf:weblog` から Weblog の URI にアクセスし、この URI に含まれる RSS の URI を取得した後に RSS にアクセスするというプロセスは煩雑であるとともに、ネットワークのトラフィックが増加するなどの問題点もある。

そこで、本研究では FOAF と RSS を接続する新たな語彙 `semblog:rss` を定義し、この要素を用いて FOAF 内に RSS の URI を記述することとする。

拡張された FOAF の生成と管理は、マネジメント層とアグリゲーション層を担うアプリケーション「RNA」(後述)によって行う。

6.5 Weblog ツールの拡張

PermaRSS は本研究における独自拡張であるため、既存の Weblog ツールでは生成することができない。しかしながら、一部の Weblog ではテンプレートとプラグインを利用して自由にコンテンツを記述、配信することが可能である。本研究では、代表的な Weblog ツールである Movable Type について PermaRSS を生成するテンプレートを定義し、普及を進めている。

6.6 FOAF TrackBack

FOAF の仕様では、ユーザ A とユーザ B が友人関係にあるとは、両者が持つ FOAF ファイルに `A knows B`、`B knows A` という形の相互リンクが記述されている状態であると定義されている。各ユーザが分散的に FOAF ファイルを管理する方法では、この相互リンクを記述するためのコストが大きく、このことが FOAF の普及を妨げているといえる。そこで、本研究では FOAF を適切に管理するためのアプリケーションである「FOAF TrackBack」を実装し、後述の「RNA」のアドインとして提供する。

FOAF TrackBack は、既存の Weblog ツールの多くが対応している TrackBack と同様の、逆リンクを生成するためのアプリケーションである。ただし、Weblog ツールの TrackBack と

は異なり，FOAF TrackBack ではツールが自動的に逆リンクを生成するのではなく，一度相手ユーザの承認を得る必要がある．これは，一方のユーザからの Spam や，プライバシーの侵害を防止するためである．

ユーザ A と B の間で友人関係を結ぶ場合の手順を以下に示す．

1. ユーザ A は，自身が持つ FOAF TrackBack アプリケーションを用いて，ユーザ B の FOAF を登録する．
2. ユーザ A の FOAF TrackBack アプリケーション（以下 FOAF TrackBack A と表記）は，ユーザ A の持つ FOAF ファイルに A knows B の情報を書き込む．
3. 2 と同時に FOAF TrackBack A はユーザ B の FOAF ファイルにアクセスし，ユーザ B が FOAF TrackBack アプリケーションを保持しているかどうかを判別する．
4. FOAF TrackBack B が存在する場合，FOAF TrackBack A は XML-RPC プロトコルを用いて FOAF TrackBack B にアクセスし，ユーザ A の FOAF 情報を送信する．
5. FOAF TrackBack B はメッセージを受け，自身の持つファイルに B isKnownBy A の情報を書き込む．
6. ユーザ B が FOAF TrackBack B にアクセスした際に，ユーザ A から受けた B isKnown A の情報を提示し，同時に B Knows A リンクを張るかどうかを決定するボタンを表示する．
7. ユーザ B がこのボタンを押した場合には，FOAF TrackBack B は FOAF TrackBack A と XML-RPC による通信を行い，ユーザ B に承認された旨を通知する．

FOAF TrackBack の機構により，ユーザは大きなコストをかけることなく FOAF による人間関係ネットワークを構築することが可能になる．

6.7 RNA: RSS 収集・配信サービス

RNA は Perl で記述された CGI プログラムである．ユーザは自身が持つ Web サーバにこれを設置して運用することができる．スクリーンショットを図 6.3 に示す．

RNA のユーザは最初に RSS の登録を行う必要がある．他サイトが配信している RSS の URI を設定すると，RNA は HTTP 通信によってファイルを取得する．登録サイトには分類のためにカテゴリを設定することができる．登録サイトのリストは RSS 化され，他のアプリケーションで使用することができる．また，アグリゲータのサイトリストの標準フォーマットである OPML の読み込み，書き出しにも対応している．

RNA は登録された RSS を取得後，パース処理を行い，複数の RSS ツリーから 1 つの

「global」RSS ツリーを構築する。global RSS ツリーは取得された全ての情報が格納されている。次に、RNA はコントローラの要求に応じて global ツリーを加工し、部分ツリーを生成する。ここでは、サイトごとの最新記事を抽出したもの、サイトにかかわらず更新时间順にコンテンツを並べるものといった3種類のツリーを生成する。また、ユーザはルールを記述したプラグインスクリプトを用意することで自由に部分ツリーを生成することができる。

生成された部分ツリーは、そのまま新しい RSS として配信するほか、XSL スタイルシートを用いて Web ブラウザ側もしくはサーバ側の XSLT エンジンによって可視化することが可能である。また、RNA 内部の HTML 変換エンジンによって、ユーザがテンプレートファイルを用意することで部分ツリーを HTML 化することも可能である。ここで用いられるテンプレートは HTML と類似したものになっており、XSL スタイルシートよりも理解しやすく一般ユーザにもカスタマイズしやすいものになっている。

RNA で表示するコンテンツのうち、ユーザが興味を持ったものに対しては、1 クリックでクリップリストに登録することができる。クリップされたコンテンツは独自の RSS ツリーに格納され、その他の RSS と同様に配信される。通常のツリーは内容が刻々と変化していくが、クリップのツリーからは情報が消されることはない。

RSS の内容は配信元の情報更新によって変化するために、RSS 取得およびツリー生成はたびたび行う必要がある。RNA はシンプルな CGI であることから、更新処理はサーバ側の cron によって一定間隔で行うことが可能である。また、閲覧者が手動で更新要求を出せるほかに、Weblog ツールが標準的に備えている XML-RPC インターフェイスのメッセージを受信し、遠隔更新することも可能である。これにより、Weblog の書き手が新たなコンテンツを作成したことを RNA に通知することができる。

RNA は取得したコンテンツのそれぞれについて後述の TrackBack リンクの有無をシステムに問い合わせ、存在する場合にはこれを抽出する。また、Description 内に記述されているハイパーリンクを同様に抽出する。抽出されたリンク情報は新たなメタデータとして配信時に追加される。

RNA では取得された RSS の文法をチェックし、正しくないものがあればこれを修正する。RSS には主に 0.91, 1.0 および 2.0 の3種のバージョンが提案されており、サイトごとに配信されている RSS のバージョンが異なっているが、RNA では全てのフォーマットをバージョン 1.0 に変換する。全ての RSS はキャッシュされるとともに、サイトごとにしか付加されていなかった RSS のツリーを分解し、コンテンツごとに付加したものを保存する。この RSS には先ほど述べたリンク情報をメタデータとして追加する。

特定のサイトを登録し、更新状況をチェックするためのアプリケーションとしてはアンテナがすでに存在する。既存のアンテナは URI に対して HTTP 通信を用いてヘッダ (LastModified

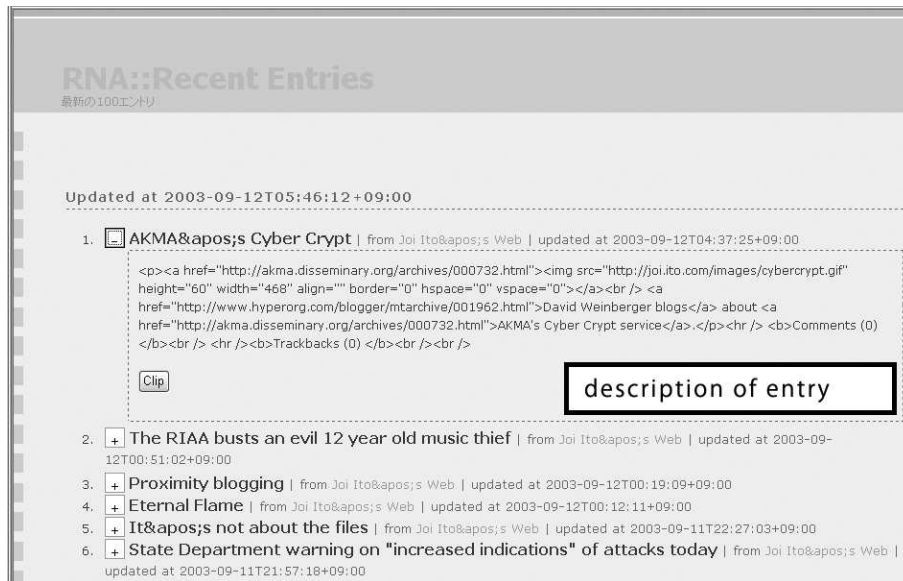


図 6.3 RNA: Snapshot

等)を取得し、それに応じてリストを並べ替える。RNA では、RSS をデータ交換フォーマットとして用いることにより、コンテンツのタイトルや作成者、概要といったリッチなコンテンツを取得できるため、RNA が配信するコンテンツもそれにともなって充実させることができる。ただし、RSS を配信していないサイトに対応するために、RNA では既存のアンテナと連携して情報を得る機能がある。

RNA では RSS に対応することでサイト単位だけでなく、サイト内のコンテンツ単位での情報編集を行うことが可能になった。クリップ機能や TrackBack 抽出機能はその一部である。また、RNA が独自に RSS を付加することにより、より充実したメタデータの配信が可能になっている。

6.7.1 glucose: クライアント型 RSS アグリゲータ

glucose は Windows PC 上で動作するクライアント型 RSS アグリゲータである。既存のクライアント型アグリゲータと異なり、glucose では RNA との連携によって情報の流通プロセスを支援することを目指して開発されている。スクリーンショットを図 6.4 に示す。

ユーザは RNA と同様に他サイトが配信する RSS の URI を登録する。OPML の入出力にも対応する。また、RSS を配信していないいくつかのニュースサイトについてはセンサープラグインという Python スクリプトによって記事を切り出し、RSS 化することが可能である。

glucose によって取得された RSS は展開され、3 ペインのインターフェイスによって表示される。左ペインは RSS を配信するサイトのリスト（チャンネル）である。右上のペインには各コンテンツのタイトル、更新日時、サイト名等のリストが表示されており、各項目によってソートすることが可能である。右下のペインには選択されたコンテンツの内容が表示される。また、ティッカー（電光掲示板）機能により、ユーザに対してプッシュ形式で情報を伝えることも可能である。

RNA と同様に各コンテンツについて TrackBack を抽出することが可能である。抽出されたリンクは右上のペインでメーラの「Re:」表示と同じように表示される。また、リンク先のコンテンツは glucose が先読みすることで、快適に閲覧することができる。

興味のあるコンテンツについてユーザ自身の Weblog に記事を追加する場合には、glucose の Weblog インターフェイスを用いて直接ポストすることができる。このインターフェイスには XML-RPC を利用している。

Weblog へのポスト機能と同様に、ユーザの持つ RNA のクリップに情報を追加することができる。

glucose は P2P のサーバントとして動作し、glucose を起動しているユーザ同士でネットワークを構成する。P2P ネットワーク内では閲覧されたコンテンツの URI 情報がランダムに配信され、他のユーザの glucose に対してコンテンツ推薦を行う。その中に興味のあるコンテンツがある場合には、クリップやポストによって反映させるほか、そのサイトの RSS を glucose に登録することができる。

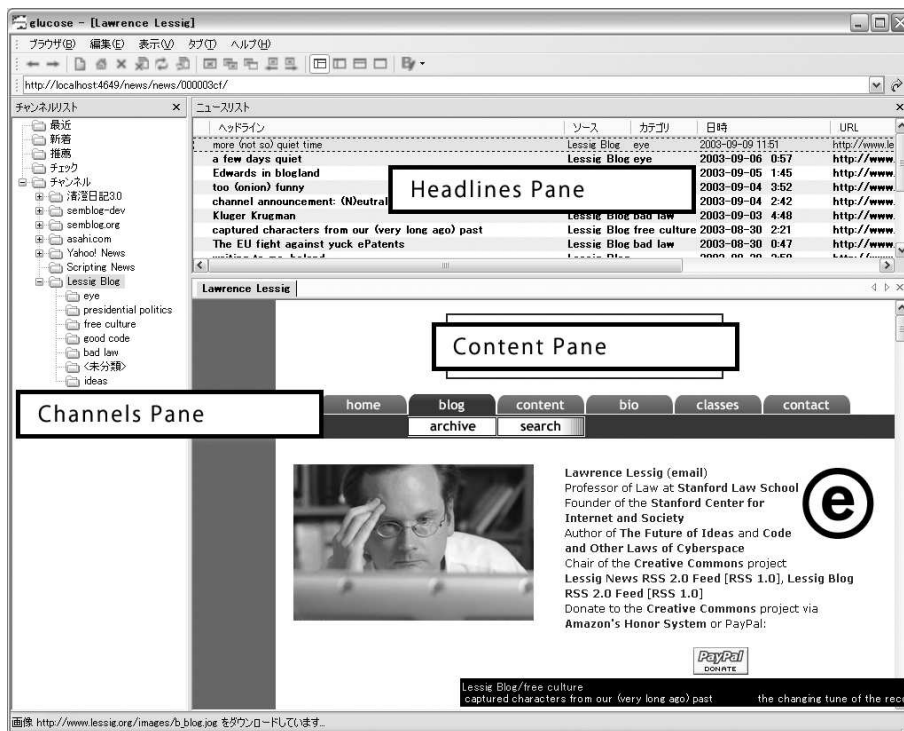


図 6.4 Glucose: Snapshot

6.8 RNA Alliance

RNA では XML-RPC プロトコルによって格納された情報の入出力が可能である．これを利用して，複数の RNA の連携による情報推薦を行うことを考える．

RNA Alliance アルゴリズムでは，RNA を用いて構築した FOAF ネットワークを利用して，個々の RNA に登録されているサイトもしくはクリップの違いに基づく情報推薦を行う．以下に手順を述べる．まず，自身の持つ RNA R_0 と，パーソナルネットワーク上の RNA R_1, \dots, R_n との類似度 S_i を以下の式で求める．

$$S_i = \frac{C_i}{N_0 + N_i}$$

ここで N_i は R_i に含まれるサイト数を示し， C_i は R_0 と R_i に共通なサイト数を示す．個々の RNA はサイトの URI リスト $R_i = \{u_0, \dots, u_k\}$ を持つ．システムは，これらの URI に対して推薦スコア $V(u)$ を以下の式で与える．

$$V_i(u) = \begin{cases} S_i & \text{if } u \in R_i \\ 0 & \text{if } u \notin R_i \quad (i = 1, \dots, n) \end{cases}$$

$$V(u) = \frac{\sum_{i=1}^n V_i(u)}{n}$$

u_i が R_0 に含まれていない場合には，システムは URI のリストをこのスコア順にソートしたうえでユーザに提示する．ユーザはこれらのサイトを 1 クリックで自身の RNA に登録することができる．また，クリップされたコンテンツの RSS ツリーに対しても同様の手法を適用することが可能である．

6.8.1 エゴセントリック検索

エゴセントリックネットワーク

Weblog における文書作成の支援として，作成者の周囲の人およびコンテンツのネットワークを利用し，作成中の文書に関連する他の文書を，検索および提示することは有用であると思われる．

本研究では，ユーザを取り巻く人間関係およびコンテンツ間の関係を利用した「エゴセントリック（自分中心）」な情報検索を提案する．エゴセントリック検索とは，「自分を中心とした

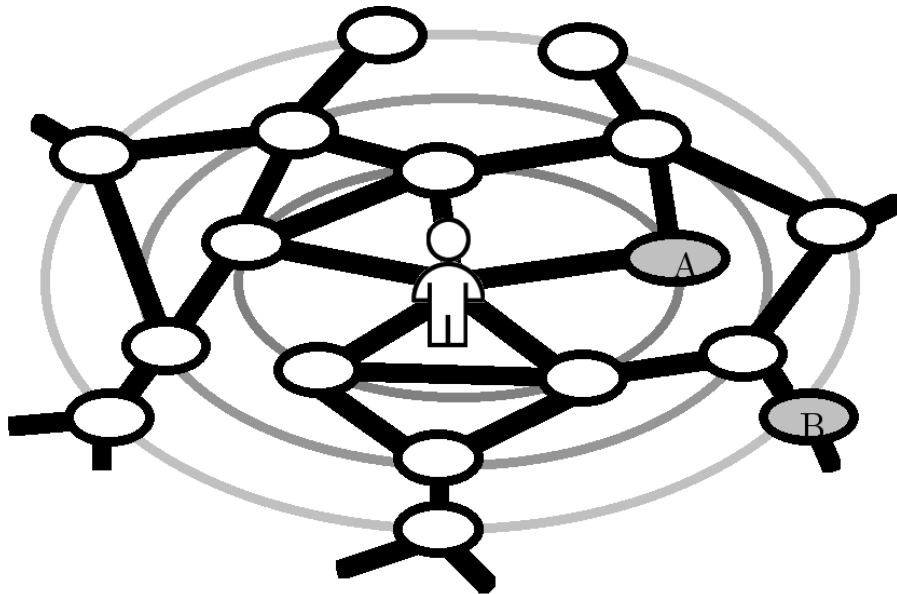


図 6.5 エゴセントリックネットワークの例

ネットワークにおいて、自分からの距離の近さに基づき情報の重要度を評価する情報検索」と定義される [71]。これは、ユーザの近くにあるコンテンツは、そのユーザにとって興味深い情報を含んでいるとの仮説に基づく。

エゴセントリック検索を用いた場合、例えば、図 6.5 のようなエゴセントリックネットワークがあったとき、自分との接続関係以外の条件がすべて同じであれば、ノード A はノード B より近くに存在するため、高く評価される。

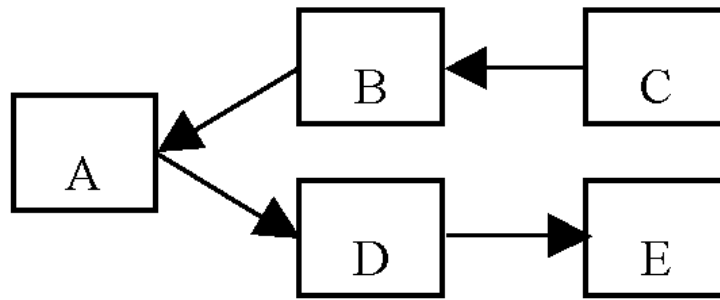
関連文書の検索には、リンク検索およびキーワード検索を用いる。リンク検索は、ユーザが作成している文書に含まれるリンクを利用した文書検索である。編集集中の文書からのリンクおよび逆リンクをもとに以下の 3 種類のコンテンツを検索する。

- a) 直接関係コンテンツ

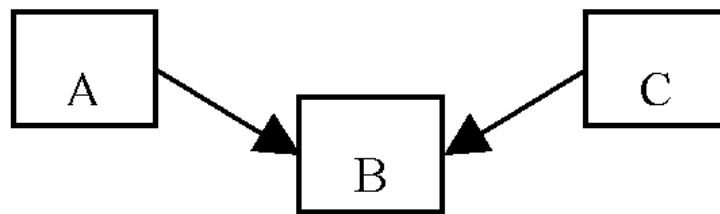
ユーザが作成している文書が参照する文書からさらに参照される文書コンテンツ、または作成中の文書を参照している文書をさらに参照している文書コンテンツを、直接関係コンテンツと呼ぶ。直接関係コンテンツによる検索手法を、Relative Chain Search という。

- b) 共参照コンテンツ

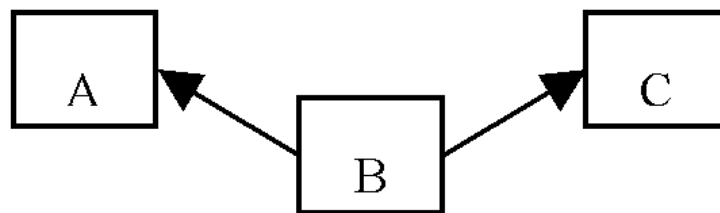
ユーザが作成している文書が参照する文書を参照している文書コンテンツを、共参照コンテンツと呼ぶ。共参照コンテンツによる検索手法を、Relative Co-reference Search



(a) Relative Chain Search



(b) Relative Co-reference Search



(c) Relative Co-citation Search

図 6.6 エゴセントリック検索

という。

- c) 共引用コンテンツ

ユーザが作成している文書を参照している文書から同時に参照されている文書コンテンツを、共引用コンテンツと呼ぶ。共引用コンテンツによる検索手法を、Relative Co-citation Search という。

図 6.6 (a) における、文書 A の直接関係コンテンツは、文書 C および E である。図 6.6 (b) における、文書 A の共参照関係コンテンツは、文書 C である。また、図 6.6 (c) における、文書 A の共引用関係コンテンツは、文書 C である。

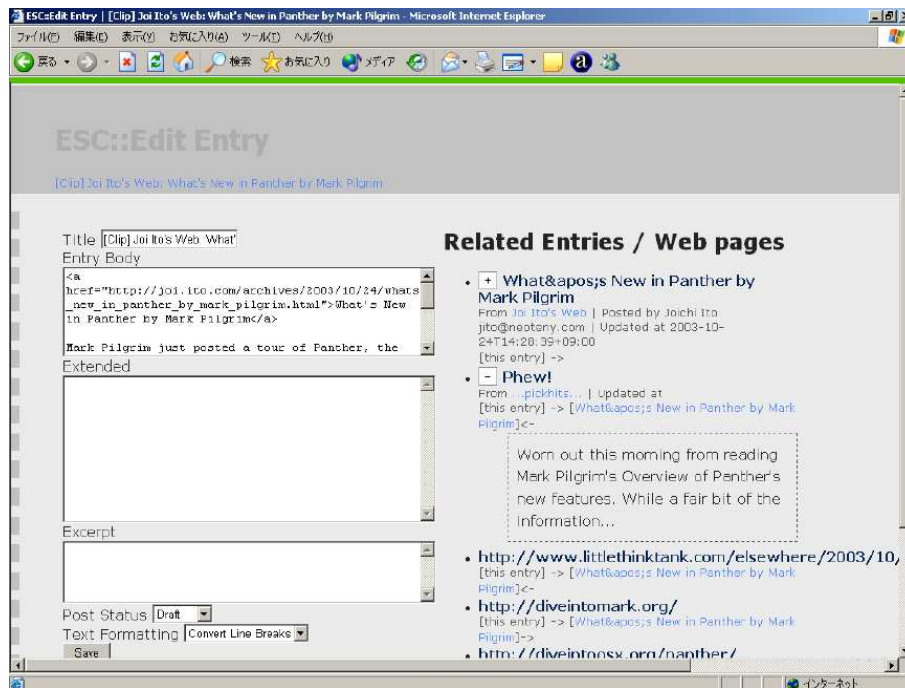


図 6.7 システム動作画面

キーワード検索は、文書間の接続関係ではなく、文書コンテンツそのものを利用する検索である。キーワード検索では、ユーザ自身が指定した検索語を含む文書を、エゴセントリックネットワーク内から検索する。これは通常の Web 検索と同様に、能動的かつ明示的に要求する文書を検索する機能をユーザに提供する。発見された文書は、ユーザと文書の間の距離をもとに順位付けて提示する。

図 6.7 に、システムの動作画面を示す。ユーザは提示された文書を読み、参考になるものがあればその文書へのリンクを、作成している文書に追加することができる。この作業によって作成中の文書が充実するとともに、エゴセントリックネットワークも更新され、システムの検索結果が変化する。これらのプロセスの繰り返しによって文書の質のさらなるブラッシュアップが期待できる。

6.9 パーソナルオントロジーの構築

スモールコンテンツを多様な形で処理するには、オントロジーを用いたセマンティックマークアップが必要不可欠である。オントロジーの構築については様々な手法が提案されているが、精密なオントロジーをトップダウンに構築するためには、専門家の知識が必要であるとともに、それらの知識を矛盾なく組織化するためのコストが非常に大きくなる。本研究では、日常的な分類行為のうち個人の知識体系が表出するとの考えから、そういった知識体系同士の連携という形でグローバルな意味体系をボトムアップに構築することを考える。そして、これらを実現するために、RSS および FOAF を利用して個人の知識体系を記述する枠組みを提案する。図 6.8 にパーソナルオントロジーの概念図を示す。

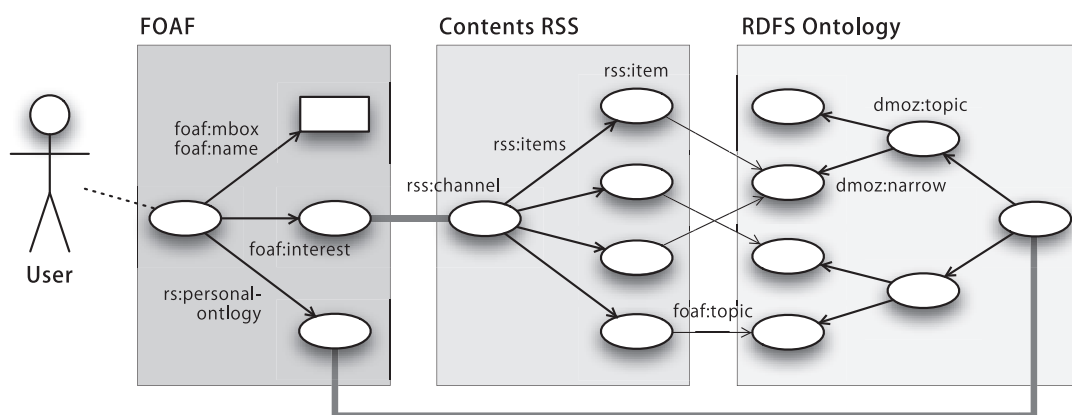


図 6.8 Personal Ontology Framework

本研究では、パーソナルオントロジーを「ツリー構造を持ったカテゴリの体系」として定義する。パーソナルオントロジーは各個人が持つものであるとし、ユーザは日常的な作業として記述もしくは収集したコンテンツをカテゴリに分類する。各カテゴリのラベルは任意である。既存のオントロジーと異なり、パーソナルオントロジーをメタデータで記述するためには、それを作成した人との関係を示す必要がある。そこで FOAF の語彙を用いて人とオントロジーの間の関連づけを行う。

パーソナルオントロジーは個人を示す FOAF、カテゴリの構造を示す RDFS オントロジー、収集および記述したコンテンツ集合を表現するコンテンツ RSS の3つから構成される。

パーソナルオントロジーで用いる FOAF には、基本的なモデルに加えて foaf:interest、semlong:personalontology の2つの要素を追加する。foaf:interest はコンテンツ RSS を示すための語彙である。semlong:personalontology は RDFS オントロジーを示すため

に、本研究において新たに定義した語彙である。この語彙は `dc:relation` のサブクラスとして定義されており、ドメインは `foaf:Agent`、レンジは `rdf:resource` である。

RDFS オントロジーの記述形式は Open Directory Project^{*1}に準ずる。各ノードにはフラグメント ID を付加する。

コンテンツ RSS は既存のものと同様に記述する。既存の RSS では `dc:subject` を用いてリテラルでカテゴリを表現する場合が多い。これに対してパーソナルオントロジーで用いる RSS では、`foaf:topic` を用いて RDFS オントロジーのフラグメント ID を指す。なお、RSS が指し示すカテゴリは必ずしもユーザ自身の持つ RDFS オントロジー内のものでなくともよく、他人の RDFS オントロジーや、その他のグローバルオントロジー内のカテゴリを示す場合もある。

このように、FOAF、コンテンツ本体およびオントロジーをそれぞれ別のファイルに分離して管理することで、既存のモデルやアプリケーションとの後方互換性を確保し、また多様な意味を表現することが可能になる。

このフレームワークによって以下のことが可能になる。ブックマークやディレクトリを対象とする 2 つのツリーの比較手法によってカテゴリ間の類似判定とインスタンス (Web ページ) のマッピングが可能になる [72]。これによって、意味的なリンクを利用したコンテンツ検索・推薦が可能になる。また、`foaf:knows` のネットワーク上で上のような検索を行い、ネットワーク距離に応じたスコアを付加することでエゴセントリック検索も容易に実現することができる [71]。

また、図 6.9 に示すように、パーソナルオントロジーと ODP、Wordnet のようなグローバルオントロジーとのマッチングをあらかじめ計算しておき、このグローバルオントロジーを介して複数のパーソナルオントロジー間の類似度計算を行うことも可能である。グローバルオントロジーはパーソナルオントロジーと同じ構造をしているため、アルゴリズムを変更する必要はない。

この手法では、それぞれのユーザはコンテンツのフォルダ分け以外に特別な作業 (アノテーションなど) を行う必要がない。また、十分な量のコンテンツが分類された後には、それを教師データとする学習手法を導入し、自動分類を行うことも可能である。そのような状態では、ユーザは一切の作業を行わずに新たなコンテンツが推薦される、いわゆるクエリーフリー検索が実現される。他にも、自らが記述したコンテンツに対して、グローバルオントロジーの分類キーワードを自動的に付加させるなど、新たなコンテンツのメタデータの表現力を高める働きも期待できる。

*1 <http://www.dmoz.org/>

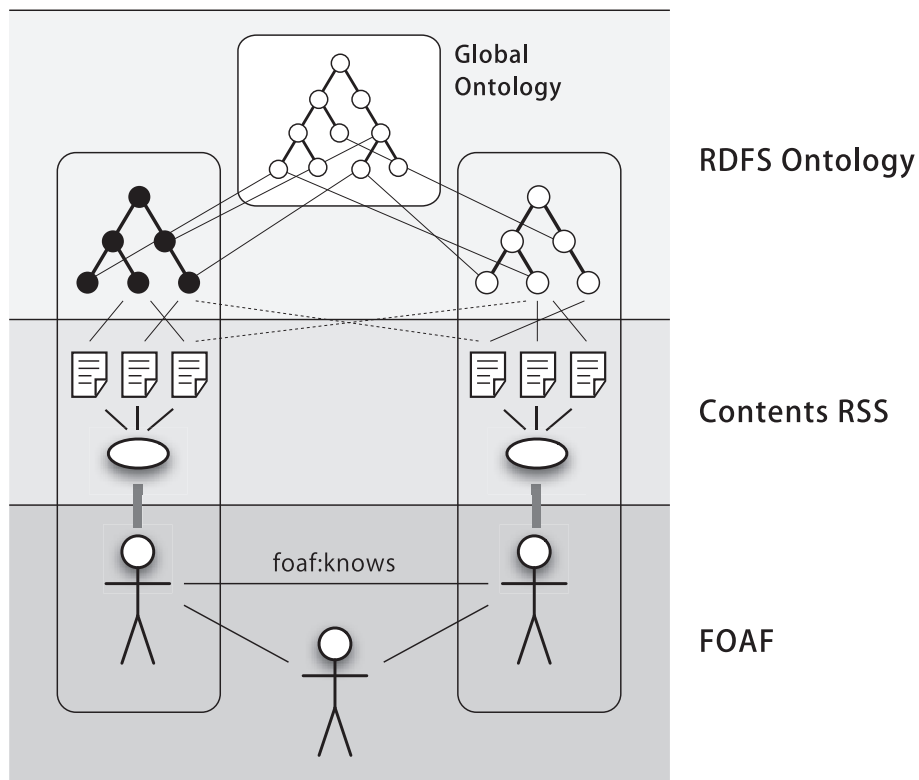


図 6.9 Bottom-up Ontology

6.10 Semblog プラットフォームを利用したコミュニティ支援

6.10.1 Weblog 情報支援サービス

Semblog プラットフォームの有効性を検証するために、実世界のコミュニティを支援するシステムを構築し、運用を行った。人工知能学会全国大会では、2003 年度よりイベント支援としてスケジュールリング支援システムをはじめとした学会支援サービスを導入している。これらのサービスは基本的に学会の開催期間という短期間に、参加者のみが利用できるかつクローズドな環境における情報支援である。2004 年度の全国大会 (JSAI2004) では、より長期間でかつオープンな環境でのコミュニケーション支援を目標として、Semblog プラットフォームを利用した情報支援サービスの提供を行った。

Semblog プラットフォームによる情報支援サービス (以下 Weblog 情報支援と称す) では、全国大会の運営側からのメッセージを Weblog 形式で配信するとともに、学会の参加者が持つ Weblog サイトを登録し、それらのサイトからの情報を集約して一覧できるようにする。また、各発表に対するコメントを参加者の Weblog を通じて書いてもらい、それを集約することで他のコメントを発表ごとに一覧することが可能である。

Weblog を用いてこのようなサービスを行うメリットは、コメントやコンテンツの管理権が書き手に残るということである。また、学会に関連するコンテンツもそうでないコンテンツも書き手の Weblog サイトで一元管理されるため、専門分野以外の情報をきっかけに新たなコミュニケーションが生まれる可能性もありうる。

JSAI2004 Weblog

JSAI2004 Weblog は全国大会の運営側からのメッセージを Weblog 形式で配信するものである。Weblog には代表的なツールである Movable Type を用いた。コンテンツは通常の Web ブラウザで閲覧することができるほか、RSS 形式での情報配信を行っているため、RSS アグリゲータなどのソフトウェアで一覧表示することも可能である。配信内容は、スタッフからのお知らせだけでなく、他のサービスと連携することで得られる情報を自動配信したものが含まれる。この自動配信内容には、人間関係ネットワーク支援サービスおよびスケジュールリング支援サービスが保持するデータより抽出された事前の人気講演リストや、会場支援システムの CoBIT センサーから得た実際の参加者の統計データがある。本サービスではこれらのデータをあらかじめ用意したテンプレート (「**さんの発表が人気です」「このセッションの聴講者は**人です」など) に適用し、サーバが定期的に XML-RPC プロトコルを用いて Weblog

ツールに書き込み処理を行う。なお，このサービスは学会の開催期間中にのみ公開された。スナップショットを図 6.10 に示す。



図 6.10 JSAI2004 Weblog

学会参加者の Weblog 一覧

このサービスでは，学会の参加者が Weblog サイトを登録し，それらのサイトからの情報を集約して時系列で一覧できるようにした。集約した情報は RSS 形式でも配信した。集約・配信ツールには RNA をカスタマイズしたものを利用した。登録される Weblog の内容は一切問わず，研究に関係のあるコンテンツもそうでないコンテンツも一元管理される Weblog の特性を生かし，インフォーマルなコミュニケーションの契機になることを目指した。スナップショットを図 6.11 に示す。

発表一覧 (Weblog 版)

本サービスでは，スケジュールリング支援システムとは別に，発表一覧を Weblog 形式で配信している。Weblog ツールには JSAI2004 Weblog と同様に Movable Type を利用した。この Weblog では 1 つの記事が 1 つの発表を示しており，それぞれの記事は TrackBack を受けつける設定になっている。各発表に対するコメントを参加者の Weblog を通じて書き，TrackBack

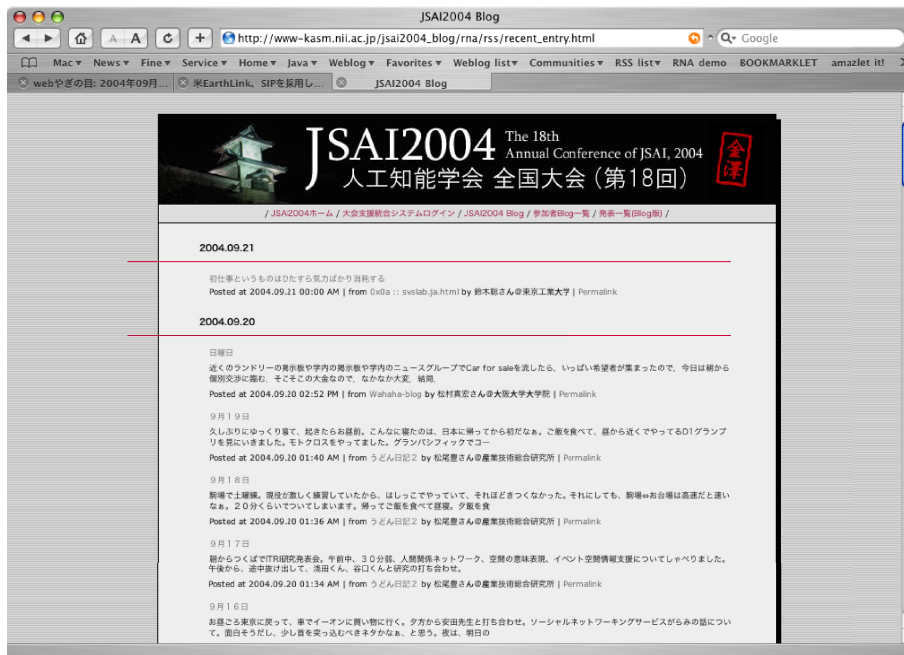


図 6.11 学会参加者の Weblog 一覧

リンクを設定することで、他のコメントとともに情報を集約し、一覧表示することが可能になる。スナップショットを図 6.12 に示す。

コンテンツ記述支援

スケジューリング支援システムや人間関係ネットワーク支援システムとの連携により、参加者の Weblog で全国大会関係のコンテンツを書く際に、ある時間に誰が近くにいたのか、どうい発表があったのかといった情報を提示する Weblog エディタを提供した。

結果とフィードバック

構築したサービスがクローズドなサービスと比較してユーザにどのような影響を与えたのかを確認するために、学会の開催後にアンケート調査を行った。アンケートの比較対象は JSAI2004 のために構築された 7 種のクローズド型のサービスである [46]。

アンケートの回答者の中で実際に Weblog 情報支援サービスを利用したのは 14.4% であった。他サービスの利用率は 8.0% から 52.0% であった。学会の開催時期（2004 年 6 月）における Weblog の普及率および知名度を考えると、決して少ない数であるとは言えない。これらのユーザのうち、自身が Weblog サイトを保持しているのは 50.0% だった。アンケート項目

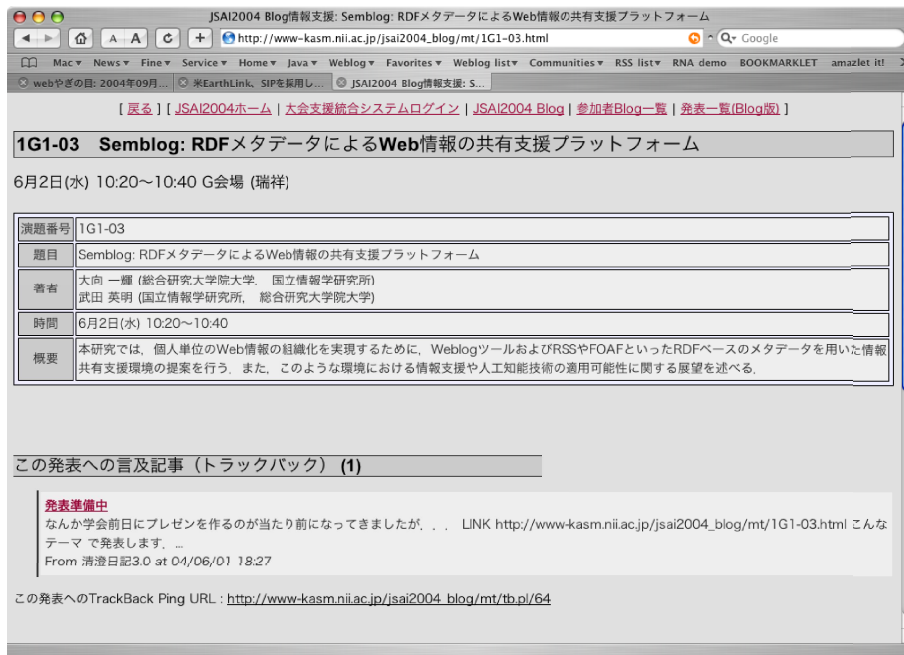


図 6.12 発表一覧 (Weblog 版)

としては、サービスの利便性を問うもの、サービス自体の面白さを問うものを基本質問として挙げた。結果は前者に対するポジティブな回答が 50.0%、後者に対するポジティブな回答が 67.5% であった。クローズドな環境でのきめ細かいサービスを行うことのできる他のサービスと比較して、利便性はやや落ちるが、サービスの結果が一般に公開される本サービスの意義について好意的な意見が見られた。学会参加者の Weblog 一覧機能について、研究関係の情報とそうでない情報を一括して集約することの是非について問う質問においては意見が分かれた。反対意見は、主にプライバシーに関する問題提起と、規模の拡大に応じて顕在化するであろう情報のノイズの問題が多かった。一方、賛成意見の中にも、集約結果をカテゴライズして表示して欲しいとの要望が多く含まれ、今後同様のサービスを運営していく上では何らかの情報分類手法を適用する必要があると思われる。

各種サービスに対するアクセス経路を知る指標として、学会参加者以外からのアクセス率を調査した。Weblog 情報支援サービスにおける外部からのアクセス率は 22.2% であった。これに対し、他のクローズド型のサービスでは 0.0% から 7.7% であったことから、Semblog プラットフォームを利用したコミュニティ支援は、クローズド型のサービスとは異なるユーザモデルを実現することが可能であることがわかった。

6.10.2 その他のコミュニティ支援

Semblog プラットフォームの技術が実際に利用されている例は数多い。本節では、コミュニティ支援の観点から RNA が利用されている例を複数取り上げる。

授業支援ブログ

Weblog がコモディティ化されていくにつれ、さまざまな Weblog ツールの利用法が提案されている。その中で、大学の授業に取り入れられるケースが増えており、トラックバック等を生かしてコンテンツの収集が行われている。しかしながら、既存のツールを単純に利用するのみでは、授業支援の目的に沿う機能が実現できない場合がある。

専修大学ネットワーク情報学部の「ウェブ日記、ウェブログによるコミュニケーション」プロジェクト^{*2}では、Semblog プラットフォームを利用した授業支援システムの構築が進められている。システムのスナップショットを図 6.13 に示す。

このシステムは、すべての学生および教師がそれぞれ Weblog サイトを持っていることを前提として、授業ごと、あるいはプロジェクトごとにメンバーの記事を集約し、表示するものである。各ユーザは、自身の Weblog に記事を投稿する際に、独自のインターフェイスを利用する。このインターフェイスには、授業やプロジェクトに関するカテゴリデータがあらかじめ埋め込まれており、ユーザがカテゴリを選択して書き込み作業を行うと、このシステムは自身の Weblog に記事を投稿するとともに、授業あるいはプロジェクトごとに設置された Weblog に対しても投稿がなされる。これにより、一次データが欠損のない形で確実に共有されることになる。

このように投稿された情報の閲覧形式としては、RNA を設置し、友人あるいは教師の Weblog の記事、もしくは彼らが興味を持った記事を RSS 化し、JavaScript を用いて自身の Weblog に組み込めるよになっている。現在は共用の RNA の情報を読み込んでいるが、今後は個人ごとに RNA を設置し、休講情報など、学生によって必要なコンテンツが異なるものを対象に情報収集を行う予定である。

組織内の個人の情報発信

個人の情報発信手段として注目されている Weblog を、Semblog プラットフォームを用いて組織内のコンテンツ管理に適用することで、コストの低い情報発信が可能になる。

^{*2} <http://www.ne.senshu-u.ac.jp/~proj16-24/>



図 6.13 授業支援システムのスナップショット

独立行政法人経済産業研究所では、公式の Web ページのコンテンツとして、組織に所属する研究員の Weblog を公開している*3。従来であれば、公式のコンテンツは単一のコンテンツ管理ポリシーに則って管理され、私的なコンテンツは各人が独自に管理し、公式ページからはリンクが張られるなどの処理が行われるのが通常である。しかしながら、前者の枠組みでは各人が自由にコンテンツを作成するにあたっての負荷が大きく、後者の方法では作成されたコンテンツが公式なものであると捉えられない恐れがある。そこで、経済産業研究所では RNA を利用して各人の Weblog の情報を集約し、公式ページにリストを埋め込むことで、両者の中間的なコンテンツ管理を実現している。図 6.14 に示すように、画面の右側には RNA が生成した各人の Weblog の記事リストが配置されている。

*3 <http://www.rieti.go.jp/it/>



図 6.14 経済産業研究所公式サイト

Weblog ユーザのコミュニティポータル

コミュニティにとって、Web における情報発信や構成員同士のコミュニケーションは非常に重要である。そのため、多くのコミュニティではポータル機能を持つサイトが構築される。しかしながら、各人が複数のコミュニティに所属しているような場合には、サイトの維持管理のコストが大きな問題となる。そこで、本研究では、Weblog と RNA を活用することで、各人の負担を軽減した上でコミュニティのポータル機能を実現することを考える。

慶應大学湘南藤沢キャンパスの学生の有志で作られたコミュニティ「TRIGGERS!」^{*4}では、参加している学生の全員が Weblog サイトを所有していることを利用して、ポータル機能の実現のために RNA を利用している。TRIGGERS!のサイトを図 6.15 に示す。RNA を用いるこ

^{*4} http://ellington.gel.sfc.keio.ac.jp/nsly/sfcrna/rss/recent_site_entry.html

とで、参加者にとっては他のメンバーの日々の活動を知ることが可能になり、外部からアクセスする人にとっては、随時コンテンツが更新されているために定期的にアクセスしやすいという利点が得られる。コミュニティの維持のためには、このような内部での継続的なコミュニケーションが必要不可欠であるため、RSS アグリゲータを利用したポータル構築は有用である。

ポータル構築のために RNA を利用した例は、「Blog 勉強会」のサイト^{*5}など数多い。



図 6.15 TRIGGERS!公式サイト

^{*5} <http://foaf.jp/research/rna/>

6.10.3 RSS アグリゲータの普及

RSS アグリゲータの普及率および認知度はともに低い*⁶。Semblog プラットフォーム、ひいては Community Web モデルが普及するためには、一般のユーザが Weblog サイトを持つとともに RSS アグリゲータを利用する必要がある。

そこで、NTT レゾナント株式会社の Web ポータル「goo」の協力により、glucose を一般ユーザ向けにカスタマイズした「goo RSS リーダー」を開発し、配布を行っている*⁷。goo RSS リーダーは glucose にインターフェイスの改善を施したものであり、基本的には機能の差異はない。スナップショットを図 6.16 に示す。

goo RSS リーダーは、2005 年 2 月現在で 100,000 を超えるダウンロード数を記録しており、国産の RSS リーダーとしては有数のユーザ数を持つ。今後、RSS による情報配信が広まるにつれ、さらなる普及が期待される。



図 6.16 goo RSS リーダー

*⁶ <http://japan.cnet.com/column/trend/story/0,2000051181,20070264-3,00.htm>

*⁷ <http://reader.goo.ne.jp/>

6.11 本章のまとめ

本章では、セマンティック Web 技術と Weblog を利用した情報流通プラットフォームについて提案を行った。RDF に基づくメタデータを普及させるために、提案システムでは Weblog ツールによってユーザに負担をかけることなく RSS や FOAF 情報を配信する。また、Web 上のコンテンツを多様化するために、“Check”、“Clip”および“Post”という3種の興味に応じた情報配信や、パーソナルオントロジーの構築を行う。提案システムは2種のRSSアグリゲータと Weblog ツール、および応用サービスから構成される。本研究で開発されたアグリゲータは100,000以上のダウンロード数を記録している。

Semblog プラットフォームの一部は、実際のコミュニティ支援システムとして構築および運用され、既存のクローズド型のサービスとは異なるユーザモデルを提供することが可能であることが確認された。

これらのシステムにより、ユーザは他の Weblog やニュースサイトから配信される RSS コンテンツを集約し、自身の新たなコンテンツ記述に生かすとともに、集約時の編集結果を容易に公開することが可能になる。編集後のコンテンツも RSS 形式で配信されることから、Community Web 活動モデルにおける Donate - Collect のプロセスは連鎖し、さらなる編集過程を経ることになる。また、FOAF を利用して構築されたパーソナルネットワーク上の他のユーザとの連携により、RNA Alliance やエゴセントリック検索、あるいはパーソナルオントロジーを用いたコンテンツ推薦を利用することが可能になる。

このような環境では、1つのトピックがユーザ間に伝播していく過程で、引用や検索など、さまざまな経路を通ることになる。ここでは、従来から言われている1次情報と2次情報の間に存在する価値の差異について、明確な定義を行うことは極めて難しい。Semblog プラットフォームの今後の課題としては、オリジナルである1次情報の価値と、コンテンツの流通に寄与した2次以降の情報の価値を分離し、どちらのコンテンツに対しても到達可能な経路を確保し、異なった観点からのスコアリングやランキングを施すことによって、両者の情報の価値を損ねないモデルを提案し、利用可能なシステムとして提供することが挙げられる。

第 7 章

結論

本論文では、実世界や Web 上での情報流通を活性化させることを目的として、個人の情報・コミュニケーション活動のモデルを定義し、これを支援するためのアプリケーションの提案、実装および検証を行った。

情報流通の過程においては、収集 (Collect)、創造 (Create)、公開 (Donete) というプロセスが存在すると考えられている。本研究ではこれらに加え、各プロセスに対応する、個人同士の交流 (Relate)、協働 (Collaborate)、発信活動 (Present) があるのではないかという仮説を提示し、コミュニケーション活動を支援することでこれまでにない情報流通支援を行うという Community Web モデルを提示した。

第3章および第4章では、このモデルを検証する第1の研究である協調的タスクスケジューリングに関して問題を概説し、この問題の解法として複数人の中での情報共有を促進するアプローチを提案した。これは前述の情報流通モデルにおいて情報配信プロセスと収集プロセスの間の連携ををどのように円滑に行うかという問題に帰着する。本研究では、これに該当するコミュニケーション活動である発信および交流プロセスを支援するための手法として、ユーザ間の過去の交流の履歴からパーソナルネットワークを構築し、このネットワーク分析を行ったうえで共有すべき情報のフィルタリングを行うシステムを提案した。これにより、過度な情報公開やプライバシーの侵害が起こる可能性を低減した上で、複数の組織に属する個人の情報公開を行うことが可能になった。これらの手法はシステムによって自動的に処理されるため、ユーザの情報入力に関わるコストを低減することにも成功した。本論文では、被験者の協力による実証実験を行い、提案手法の有効性を確認した。この研究により、Community Web モデルの実現に際する課題の1つである、Donate - Collect 間のループ形成に際する適切な情報流通が可能になった。

第5章および第6章では、情報・コミュニケーション活動におけるすべてのプロセスを支援することを目標として、Web 上でのコンテンツ流通環境の提案および設計、実装を行った。Web 上で個人がコンテンツを作成し、公開に至るまでには、事前に情報収集を行うことが珍しくない。あるユーザが公開したコンテンツは、別のユーザの収集対象になることから、第1の研究と同じく情報・コミュニケーション活動のモデルがあてはまる。そして、本研究では、ユーザがどの情報を収集するかという判断は、その情報の作り手の信頼性に基づくという仮説から、個人単位でのコンテンツの組織化、ならびに個人同士のパーソナルネットワークを利用した情報流通環境の提案を行った。これを実現するためには、適切なコンテンツマネジメントと、コンテンツの定型化、および個人間の関係の明示化が必要となるが、本研究では近年急速普及が進んでいる Weblog の諸技術ならびにセマンティック Web の諸技術を利用することで問題の解決を目指した。具体的には、現存の技術を4つのレイヤーに分類し、それぞれの要素技術についてデータフォーマットの拡張や新規アプリケーションの提案、実装を行った。低

レベルのレイヤーではコンテンツに関するメタデータとパーソナルネットワークに関するメタデータを融合させるモデルを提案し、これを既存のツールで適切に管理するシステムを構築した。次に、他のユーザが配信するメタデータを集約するためのアグリゲータを設計、実装し、ユーザが多様な興味を簡単な操作で Web 上に公開できるようにした。そして、流通したコンテンツを効果的に収集するための検索アルゴリズムおよび推薦アルゴリズムを提案し、これを実装した。そして、提案モデルを検証するためにコミュニティ支援の現場で実運用を行い、有効性を確認した。

以上の研究成果により、情報流通とコミュニケーションプロセスの統合モデル、そしてパーソナルネットワークを利用した支援モデルは有効に機能することが確かめられた。今後は、他の問題領域に同様のモデルおよびプラットフォームを適用する研究や、Web における情報流通の問題をさらに詳しく検討することで、情報流通の主体である人間とそのネットワークがどのように評価されるべきかについて考察を深める予定である。とくにセマンティック Web が目的としている「Trust」の概念 [44] が、コミュニティ支援などの研究分野においてどのように位置づけられるかに興味を持っている。

本研究は情報共有や社会ネットワーク、セマンティック Web などさまざまな研究分野の成果を統合したものである。その中で、本研究の独自性はプロトタイプではない実運用可能なシステムを構築し、すでに実運用の段階に達していることにある。また、研究の成果を社会に問うためには研究者自身が積極的に活動すべきであるという信念から、本研究のすべての成果は公開、配布あるいは実際のサービスとして運用しており、高い評価を得ている。今後は、このプラットフォーム上で新たな研究を進めることが目標であるが、それ以外にも他の研究者が自身の手法を適用できる環境として利用されることを願う。

謝辞

本研究は、多くの人々の協力や助言なくしては遂行することができませんでした。本論文を締めくくるにあたり、お世話になった方々への感謝の気持ちを以下に述べます。

まず、総合研究大学院大学複合科学研究科情報学専攻における3年間の研究生活を支えていただいた、国立情報学研究所の武田英明教授に心より感謝を申し上げます。武田教授には、研究テーマが明確に定まらない状態で博士課程への進学を認めていただき、入学後は研究指導や的確なアドバイスのみならず、研究者としての心がけについてご指導いただきました。その一方で、未熟な私を一人前の研究者として扱っていただき、研究テーマの選定や研究の遂行方法については私の考えを最大限に認めていただきました。新たなテーマに対して、それを一過性のものとは考えず、そこに内在する普遍的な知見を得るために積極的に取り組むべきであるという武田教授の先進的かつ柔軟な思考には励まされ、またこのような思考をいつも忘れてはならないとの思いを強くしました。本当にありがとうございました。

また、国立情報学研究所の上野晴樹教授ならびに山田誠二教授には、主任指導教官として総合研究大学院大学および国立情報学研究所にて研究生活を送るための支援をいただきました。研究を始める際、また本論文を執筆する際に、人工知能研究、エージェント研究の立場から多くのアドバイスをいただくことができました。ここに厚く感謝申し上げます。

本論文の審査を行っていただいた東京大学の中小路久美代特任教授、国立情報学研究所の村田剛志助教授、ならびに京都大学の角康之助教授には、本論文に対して有益なアドバイスを多数いただきました。とくに、本研究が情報学研究のどの部分に位置づけられるのかという指摘は、普段の研究生活で忘れがちな、大域的な視点の重要性を思い出させる重要なものでした。ここに厚く感謝申し上げます。

本論文に含まれる2種の研究は、どちらも情報処理推進機構による未踏ソフトウェア創造事業の支援を受けています。この事業に採択されるにあたり、提案書を評価していただき、数々のアドバイスや議論の場を提供していただいた、早稲田大学の村岡洋一教授、京都大学の石田亨教授、ならびに名古屋大学の長尾確教授に厚く感謝申し上げます。また、未踏プロジェクト

の中でさまざまな支援をしていただいたみなさま，ならびに情報処理推進機構のみなさまに深く感謝いたします。

Weblog 研究を開始するにあたっては，学部および修士時代の指導教員である同志社大学の三木光範教授ならびに廣安知之助教授，そして知的システムデザイン研究室の学生のみなさんから多くのヒントを得ることができました。両先生にはその他にも研究内容や研究生活についての相談に数多く載っていただきました。深く感謝いたします。

3年間の研究生活においては，ともに目的を同じくする研究室のメンバーの存在が必要不可欠でした。国立情報学研究所の市瀬龍太郎助手には，いつもの確かつ温かいアドバイスをいただくことができました。また，研究者として最も身近な先輩である市瀬助手には，将来についての相談にも乗っていただきました。深く感謝いたします。また，研究室では同期として，自宅では同居人として，公私を問わずこの3年間もっとも結びつきの強かった濱崎雅弘氏には心から感謝したいと思います。ある時には共同で1つの問題に取り組み，ある時には良きライバルとして成果を競い合える人間が身の回りにいたことは私にとって幸運でした。後輩である総合研究大学院大学の沼晃介君，東京工業大学の鈴木聡君，横浜国立大学の上松大輝君，ならびに位置情報プロジェクトのみなさんにも様々な意見をいただきました。感謝いたします。他にも，武田研究室との共同研究や Weblog 勉強会でお世話になった松尾豊氏をはじめとする産業技術総合研究所ならびに東京大学のみなさんとは有益な議論を数多く行うことができました。深く感謝いたします。

本研究で提案したアプリケーションを開発する際には，多くの友人の協力や貢献がありました。携帯電話用スケジューラに関わっていただいた日本電気株式会社の赤塚浩太氏，RSSリーダーの開発をリードしていただいた早稲田大学の安達真氏ならびに日本アイ・ピー・エム株式会社の佐野正樹氏に深く感謝いたします。

他にも，総合研究大学院大学複合科学研究科情報学専攻の1期生としてともに学び，議論してきた同期のみなさん，未踏ソフトウェア創造事業での出会いから多くの有能な方々にお会いするきっかけを作ってくくださった東京大学・PICSY プロジェクトの鈴木健氏やその友人のみなさん，時に専門外の視点から意見をいただいた特定非営利活動法人コラボリズムの友人たち，本研究の成果に注目していただき，共同で商品化を進めてきたエヌ・ティ・ティ・レゾナント株式会社の関係者のみなさんなど，非常に多くの方々の支援を得ることができました。心から感謝したいと思います。

最後に，京都から私の研究生活を見守ってくれた家族に心から感謝します。本当にありがとう。

参考文献

- [1] 財団法人インターネット協会（編）. インターネット白書 2004. インプレス, 2004.
- [2] B.Shneiderman. *Leonardo's Laptop: Human Needs and the New Computing Technologies*. MIT Press, 2002.
- [3] Herbert A. Simon. *The New Science of Management Decision*. Prentice-Hall, 1977.
- [4] 吉田匡志, 伊藤雄介, 沼尾正行. 口コミによる分散型情報収集システム. マルチ・エージェントと協調計算ワークショップ (MACC2001), 2001.
- [5] 小野智弘, 西山智, 堀内浩規. エージェントのための私的信用情報を利用した仲介機構の提案. 情報処理学会研究報告, 2001.
- [6] Lawrence Lessig. *The Future of Ideas: The Fate of the Commons in a Connected World*. Random House, 2001.
- [7] 黒田充, 村松健児. 生産スケジューリング. 朝倉書店, 2002.
- [8] 久保幹雄. 運搬スケジューリング問題とその周辺. 日本オペレーションズリサーチ学会シンポジウム, 1999.
- [9] 手塚大, 樋地正浩. 実用的なジョブショップスケジューリング問題のための新しい遺伝表現とコモンクラスタ交叉. 情報処理学会論文誌, Vol. 42, No. 9, pp. 2284–2292, 2001.
- [10] 電気学会 GA 等組合せ最適化手法応用調査専門委員会. 遺伝アルゴリズムとニューラルネット. コロナ社, 1998.
- [11] 石井裕. CSCW とグループウェア. オーム社, 1994.
- [12] 垂水浩幸. グループウェアとその応用. 共立出版, 2000.
- [13] 石井裕. グループウェアのデザイン. 共立出版, 1994.
- [14] 宇井徹雄. 意思決定支援とグループウェア. 共立出版, 1995.
- [15] L.Garrido and K.Sycara. Multi-Agent Meeting Scheduling: Preliminary Experiment Results. *Proceedings of Second International Conference on Multi-Agent Systems (ICMAS96)*, pp. 95–102, 1996.

- [16] T.Haynes, S.Sen, N.Arora, and R.Nadella. An Automated Meeting Scheduling System that Utilize User Preference. *Proceedings of the First International Conference on Autonomous Agents (Agents97)*, pp. 308–315, 1997.
- [17] Michael P. Wellman, Jeffrey K. MacKie-Mason, and Daniel M. Reeves. Exploring Bidding Strategies for Market-Based Scheduling. *Proceedings of the 4th ACM Conference on Electronic Commerce*, 2002.
- [18] 伊藤孝行, 新谷虎松. 好みに基づく分散 ATMS を用いたグループスケジュール管理システムについて. 人工知能学会研究報告, 1998.
- [19] 暦本純一. Time-machine computing : 時間志向ユーザインタフェースの提案. *Workshop on Interactive Systems and Software*, 1999.
- [20] 乃村能成, 花田泰紀, 牛島和夫. MHC - Message Harmonized Calendaring System の設計と実装. 情報処理学会論文誌, Vol. 42, No. 10, pp. 2518–2525, 2001.
- [21] R.Blood. *The Weblog Handbook*. Perseus Publishing, 2002.
- [22] E.Aimeur, G.Brassard, and S.Paquet. Using Personal Knowledge Publishing to Facilitate Sharing Across Communities. *Workshop on (Virtual) Community Informatics, Held in conjunction with the Twelfth International World Wide Web Conference (WWW2003)*, 2003.
- [23] R.Kumar, J.Novak, P.Raghavan, and A.Tomkins. On the Bursty Evolution of Blogspace. *Proceedings of the Thirteenth International World Wide Web Conference (WWW2003)*, 2003.
- [24] HP Laboratory. Blog Epidemic Analyzer. <http://wwwid1.hp1.hp.com/blogstuff/>.
- [25] E.Adar, L.Zhang, L.A.Adamic, and R.M.Lukose. Implicit Structure and the Dynamics of Blogspace. *WWW2004 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, 2004.
- [26] B.A.Nardi, D.J.Schiano, M.Gumbrecht, and L.Swartz. "I'm Blogging This" A Closer Look at Why People Blog. *Communications of ACM*, 2004.
- [27] D.J.Schiano, B.A.Nardi, M.Gumbrecht, and L.Swartz. Blogging by the Rest of Us. *Proceedings of the CHI2004*, 2004.
- [28] S.C.Herring, L.A.Scheidt, and E.W.Bonus. Bridging the Gap: A Genre Analysis of Weblogs. *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS04)*, 2004.
- [29] K.Gill. How can we measure the influence of the blogosphere? *WWW2004 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, 2004.

- [30] WWW2004 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics. <http://www.blogpulse.com/www2004-workshop.html>, 2004.
- [31] 日記を認めるシステムたち. <http://kohgushi.fastwave.gr.jp/aboutdiary/>.
- [32] 南野朋之, 鈴木泰裕, 藤木稔明, 奥村学. blogの自動収集と監視. 人工知能学会論文誌, Vol. 19, No. 6, pp. 511–520, 2004.
- [33] Y.Kawaura, Y.Kawakami, and K.Yamashita. Keeping a diary in cyberspace. *Japanese Psychological Research*, Vol. 40, No. 4, pp. 234–245, 1998.
- [34] 川浦康至, 山下清美, 川上善郎. 人はなぜウェブ日記を書き続けるのか. 社会心理学研究, Vol. 14, pp. 133–143, 1999.
- [35] 松村真宏, 三浦麻子, 柴内康文, 大澤幸生, 石塚満. 2ちゃんねるが盛り上がるダイナミズム. 情報処理学会論文誌, Vol. 45, No. 3, pp. 1053–1061, 2004.
- [36] T.Berners-Lee. A roadmap to the Semantic Web. <http://www.w3.org/DesignIssues/Semantic.html>, 1998.
- [37] T.Berners-Lee, J.Hendler, and O.Lassila. The Semantic Web. *Scientific American*, 2000.
- [38] World Wide Web Consortium (W3C). OWL Web Ontology Language Overview. <http://www.w3.org/TR/owl-features/>, 2003.
- [39] 橋田浩一. GDA: 意味的修飾に基づく多用途の知的コンテンツ. 人工知能学会誌, Vol. 13, No. 4, pp. 528–535, 1998.
- [40] S.Dill, N.Eiron, D.Gibson, and et al. SemTag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation. *Proceedings of the Twelfth International World Wide Web Conference (WWW2003)*, 2003.
- [41] 岩爪道昭, 白神謙吾, 畑谷和右, 武田英明, 西田豊明. オントロジーに基づく広域ネットワークからの情報収集・分類・統合化. 情報処理学会論文誌, Vol. 38, No. 3, pp. 606–615, 1997.
- [42] 溝口理一郎, 池田満. オントロジー工学序説. 人工知能学会誌, Vol. 12, No. 4, pp. 559–569, 1997.
- [43] 市瀬龍太郎, 濱崎雅弘, 武田英明. 階層的な分類データを統合するための規則学習機構. 人工知能学会論文誌, Vol. 19, No. 6, 2004.
- [44] J.Golbeck and J.Hendler. Accuracy of Metrics for Inferring Trust and Reputation in Semantic Web-based Social Networks. *Proceedings of the 14th International Conference on Knowledge Engineering and Knowledge Management (EKAW04)*, 2004.
- [45] L.A.Adamic and E.Adar. Friends and Neighbors on the Web. *Social Networks*, 2002.

- [46] 濱崎雅弘, 武田英明, 大向一輝, 市瀬龍太郎. パーソナルネットワークを利用したコミュニティシステムの提案と分析. *人工知能学会論文誌*, Vol. 19, No. 5, pp. 389–398, 2004.
- [47] Joshua R. Tyler, Dennis M. Wilkinson, and Bernardo A. Huberman. Email as Spectroscopy: Automated Discovery of Community Structure within Organizations. *Condensed Matter*, 2003.
- [48] Noshir S. Contractor, Daniel Zink, and Michael Chan. A Tool to Assist and Study the Creation, Maintenance, and Dissolution of Knowledge Networks. *Community Computing and Support Systems*, pp. 201–217, 1998.
- [49] Y.Matsuo, H.Tomobe, K.Hasida, and M.Ishizuka. Finding Social Network for Trust Calculation. *Proc. 16th European Conf. on Artificial Intelligence (ECAI2004)*, pp. 510–514, 2004.
- [50] D.J.Watts. *Small Worlds: The Dynamics of Networks Between Order and Randomness*. Princeton Univ, 2004.
- [51] A.Barabasi. *Linked: The New Science of Networks*. Perseus Books Group, 2002.
- [52] 安田雪. 人脈づくりの科学. 日本経済新聞社, 2004.
- [53] N.Yoshioka, Y.Tahara, and S.Honiden. Active Contents: Flexible Contents Distribution by Mobile Agents. *IPSJ Transactions on Databases*, Vol. 44, No. SIG18-006, 2003.
- [54] Matthias Klusch. Information Agent Technology for the Internet: A survey. *Data and Knowledge Engineering*, Vol. 36, , 2001.
- [55] Clarence A. Ellis, Simon J. Gibbs, and Gail Rein. Groupware: Some Issues and Experiences. *Communications of the ACM*, Vol. 34, No. 1, pp. 39–58, 1991.
- [56] Barry Wellman. Computer Networks As Social Networks. *Science*, Vol. 293, pp. 2031–2034, 2001.
- [57] I. Ohmukai, H. Takeda, and M. Miki. A Proposal of the Person-centered Approach for Personal Task Management. *Proceedings of 2003 Symposium on Applications and the Internet (SAINT2003)*, pp. 234–240, 2003.
- [58] S. J. Noronha and V. V. S. Sarma. Knowledge-Based Approaches for Scheduling Problems: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 3, No. 2, pp. 160–171, 1991.
- [59] 亀井剛次, 藤田邦彦, E. Jettmar, 吉田仙, 桑原和宏. ネットワークコミュニティの形成を支援するシステム Community Organizer における情報提示手法の検討. *電子情報通信学会論文誌*, Vol. J84-D-I, No. 9, pp. 1440–1449, 2001.

- [60] 工藤祐子, 大谷信介. 数量として測る友人数. <http://www-soc.kwansei.ac.jp/otani/research/02sotsuron/3-3.htm>, 2000.
- [61] 富田悦次, 平賀直仁, 若月光夫. 最大クリーク抽出アルゴリズムの効率化とその評価. 情報処理学会研究報告, Vol. 99-MPS-24, No. 1, pp. 1-4, 1999.
- [62] Sun Microsystems, <http://jcp.org/aboutJava/communityprocess/final/jsr139/>. *Java 2 Platform Micro Edition Connected Limited Device Configuration (J2ME CLDC) Specification*, 2003.
- [63] NTT DoCoMo, http://www.nttdocomo.co.jp/p_s/imode/java/pdf/jguide504_021216.pdf. *i アプリコンテンツ開発ガイド for 504i*, 2002.
- [64] B.Hammersley. *Content Syndication with RSS*. O'Reilly & Associates, 2003.
- [65] F.Manola and E.Miller. RDF Primer (W3C Recommendation). <http://www.w3.org/TR/rdf-primer/>, 2004.
- [66] D.Brickley and L.Miller. FOAF Vocabulary Specification. <http://xmlns.com/foaf/0.1/>, 2004.
- [67] Weblog ツールリスト. <http://artifact-jp.com/weblog/>.
- [68] A.Broder, R.Kumar, F.Maghouli, P.Raghavan, and R.Stata. Graph structure in the Web. *Proceedings of the 9th International World Wide Web Conference (WWW2000)*, pp. 247-256, 2000.
- [69] UserLand Software. XML-RPC Specification. <http://www.xmlrpc.com/spec>, 1999.
- [70] Six Apart. Movable Type. <http://www.movabletype.org/>, 2003.
- [71] I.Ohmukai, K.Numata, and H.Takeda. Egocentric Search Method for Authoring Support in Semantic Weblog. *Workshop on Knowledge Markup and Semantic Annotation (Semannot2003), Held in conjunction with the Second International Conference on Knowledge Capture (K-CAP2003)*, 2003.
- [72] M.Hamasaki and H.Takeda. Experimental Results for a Method to Discover Human Relationship based on WWW Bookmarks. *Proceedings of the Fifth International Conference on Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies (KES2001)*, pp. 1291-1295, 2001.

研究業績

主著論文

学術論文（査読付き）

1. 大向一輝, 武田英明. 人間関係ネットワークに基づく情報フィルタリングを用いた協調的タスクスケジューラ. 電子情報通信学会論文誌, Vol. J87-D1, No. 11, pp. 1020–1029, 2004.

国際会議

口頭発表

1. Ikki Ohmukai, Hideaki Takeda, and Mitsunori Miki. A Proposal of the Person-centered Approach for Personal Task Management. *Proceedings of 2003 Symposium on Applications and the Internet (SAINT2003)*, pp. 234–240, 2003.
2. Ikki Ohmukai and Hideaki Takeda. Social Scheduler: A Proposal of Collaborative Personal Task Management. *Proceedings of Web Intelligence (WI2003)*, 2003.
3. Ikki Ohmukai, Hideaki Takeda, Kosuke Numa, Masahiro Hamasaki, and Shin Adachi. Personal Knowledge Publishing Suite with Weblog. *Workshop on Weblogging Ecosystem, Held in conjunction with the Thirteenth International World Wide Web Conference (WWW2004)*, 2004.
4. Ikki Ohmukai, Hideaki Takeda, Kosuke Numa, Masahiro Hamasaki, and Shin Adachi. Metadata-driven Personal Knowledge Publishing. *Proceedings of the Third International Semantic Web Conference (ISWC2004)*, 2004.
5. Ikki Ohmukai and Hideaki Takeda. Collaborative Task Scheduling Method based on Social Network Analysis for Cellphone Application. *Proceedings of the IADIS*

International Conference of WWW/Internet (ICWI2004), 2004.

ポスター発表

1. Ikki Ohmukai, Kosuke Numa, and Hideaki Takeda. Egocentric Search Method for Authoring Support in Semantic Weblog. *Workshop on Knowledge Markup and Semantic Annotation (Semannot2003)*, Held in conjunction with the Second International Conference on Knowledge Capture (K-CAP2003), 2003.

全国大会・研究会

1. 大向一輝, 濱崎雅弘, 武田英明, 三木光範. Social Scheduler - P2P モデルを用いた協調的リソースプランナの提案. 第 16 回人工知能学会全国大会論文集, 2002.
2. 大向一輝, 武田英明, 三木光範. 多様かつ曖昧な個人タスクのための管理システムの提案と実装. エージェント合同シンポジウム (JAWS2002) 講演論文集, pp. 502-509, 2002.
3. 大向一輝, 武田英明. 携帯電話を用いた協調的タスクスケジューラ. 第 17 回人工知能学会全国大会論文集, 2003.
4. 大向一輝, 武田英明. 人間関係グラフに基づく情報フィルタリングを用いた協調的タスクスケジューラ. *Joint Agent Workshops & Symposium (JAWS2003)*, 2003.
5. 大向一輝, 武田英明. Semblog: RDF メタデータによる Web 情報の共有支援プラットフォーム. 第 18 回人工知能学会全国大会論文集, 2004.

主著以外の論文

学術論文（査読付き）

1. 濱崎雅弘, 武田英明, 大向一輝, 市瀬龍太郎. パーソナルネットワークを利用したコミュニティシステムの提案と分析. 人工知能学会論文誌, Vol. 19, No. 5, pp. 389–398, 2004.
2. 濱崎雅弘, 武田英明, 大向一輝, 市瀬龍太郎. 学術会議における共有型スケジューリング支援システムの開発と運用. 日本データベース学会 Letters (DBSJ Letters), Vol. 2, No. 4, pp. 7–10, 2004.

国際会議

ポスター発表

1. Masahiro Hamasaki, Hideaki Takeda, Ikki Omukai, and Ryutaro Ichise. Scheduling support system for academic conferences based on interpersonal networks. In *Demonstration and Poster Proceedings of Hypertext2004*, pp. 50–51, 2004.
2. Kosuke Numa, Ikki Ohmukai, Masahiro Hamasaki, and Hideaki Takeda. Egocentric Search based on RSS. *Proceedings of the Third International Semantic Web Conference (ISWC2004)*, 2004.
3. H.Uematsu, T.Tokunaga, Kosuke Numa, Ikki Ohmukai, and Hideaki Takeda. Ba-log: Location-based Information Aggregation System. *Proceedings of the Third International Semantic Web Conference (ISWC2004)*, 2004.

全国大会・研究会

1. 濱崎雅弘, 武田英明, 大向一輝, 市瀬龍太郎. 学術会議における共有型スケジューリング支援システムの開発と運用. データベースと Web 情報システムに関するシンポジウム (DBWeb2003), pp. 147–154, 2003.
2. 濱崎雅弘, 武田英明, 大向一輝, 市瀬龍太郎. 流通情報に基づいた自律分散最適化を行うネットワーク型情報共有システムの提案. *Joint Agent Workshops & Symposium (JAWS2003)*, 2003.

3. 濱崎雅弘, 武田英明, 大向一輝, 市瀬龍太郎. 2003 年度スケジューリング支援システムの開発と運用. 第 18 回人工知能学会全国大会論文集, 2004.
4. 上松大輝, 徳永徹郎, 沼晃介, 大向一輝, 武田英明. 場 log: 位置情報に基づいた情報整理システム. インタラクション 2004, 2004.
5. 濱崎雅弘, 武田英明, 大向一輝, 市瀬龍太郎. コミュニティシステムのためのパーソナルネットワークの利用とその分析. 第 18 回人工知能学会全国大会論文集, 2004.
6. 沼晃介, 大向一輝, 濱崎雅弘, 武田英明. Weblog における文書作成支援のためのエゴセントリック検索. 第 18 回人工知能学会全国大会論文集, 2004.
7. 沼晃介, 大向一輝, 濱崎雅弘, 武田英明. Weblog におけるエゴセントリック検索の提案と実装. 人工知能学会セマンティックウェブとオントロジー研究会, 2004.
8. 上松大輝, 沼晃介, 徳永徹郎, 大向一輝, 武田英明. 場 log: Weblog 環境における位置情報利用の提案. 人工知能学会セマンティックウェブとオントロジー研究会, 2004.

その他

著書（分担執筆）

1. 宮川達彦, 伊藤直也（編）. *Weblog Hacks*. オライリー・ジャパン, 2004.

解説記事

1. 武田英明, 大向一輝. Weblog の現在と展望 : セマンティック Web およびソーシャルネットワークワーキングの基盤として. *情報処理*, Vol. 45, No. 6, pp. 586–593, 2004.

表彰

1. 大向一輝. Semblog: セマンティックウェブ技術を用いたスモールコンテンツの再編集・共有プラットフォーム. 情報処理推進機構「平成 15 年度未踏ソフトウェア創造事業」スーパークリエイター, 2004.

付録 A

協調的タスクスケジューラのデータ構造

協調的タスクスケジューラのデータ構造を以下に示す。

個人タスクのデータ構造

サーバ上のデータベースおよび携帯端末内で保存される個人タスクのデータは以下のパラメータを持つ。

- UID : (例 : 20021221160553i2k)
個々のタスクにユニークにつけられた ID 番号 . TIMESTAMP (後述) + 乱数 + 依頼者の ID から構成されている .
- TITLE : (例 : 研究室ミーティング)
タスクの名称 . Shift-JIS で記述する .
- START : (例 : 2003/02/16 09:00)
タスクの開始時間 . YYYY/MM/DD HH:mm の形式で記述する .
- END : (例 : 2003/02/16 18:00)
タスクの終了時間もしくは締切時間 . 書式は START と同様である .
- PRIORITY : (例 : 5)
タスクの主観的な重要度 . 1~5 の整数で記述する . 5 が最重要である .
- TIMESTAMP : (例 : 2003/01/23 15:22)
タスクが発行された時間 . 書式は START と同様である .

- TODO :(例 : false)
タスクの種類判別フラグ . true であれば TODO タスク , false であれば Schedule タスクである .
- WORKLOAD :(例 : 30)
TODO タスクの仕事量 . 整数で記述する . 単位は時間である .
- ACCEPT :(例 : 21)
協調・依頼タスクにおいて他のユーザが受理しているかどうかを示すフラグ .
- SYNC :(例 : true)
タスクがサーバ上のデータベースと携帯端末内で同期されているかどうかを示すフラグ . true であれば同期済み . 情報が変更されると false になる .
- PUB :(例 : -1)
他ユーザにこのタスクを閲覧させてもよいかどうかを示すフラグ . -1 は完全非公開 , 1 は完全公開 , 0 はスマートフィルタリング適用を示す .
- GROUP :(例 : 1)
タスクの種類判別フラグ . -1 が個人タスク , 1 が依頼タスク , 0 が協調タスクを示す .
- FROM :(例 : ham)
タスクの依頼者の ID .
- MEMBER :(例 : ham:i2k:akatsuka)
タスクの実行者の ID . 協調・依頼タスクでは個々の ID をコロン区切りで記述する .
- MEMO :(例 : 国立情報学研究所にて)
タスクに関する備考 . Shift-JIS で記述する .
- PROGRESS :(例 : 15)
TODO タスクにおける進捗度 . 0 ~ 100 の整数で記述する . 単位は % である .

協調・依頼タスクのデータ構造

協調・依頼タスクのデータは各ユーザのタスクデータに中にも保持されるが , それとは別にサーバ上でも一括管理される . パラメータは以下の通りである . なお , 個人タスクデータと同様のパラメータについては説明を省略する .

- UID
- TITLE
- LOCK :(例 : true)

タスク情報を変更できるかどうかを示すフラグ。true であれば変更不可，false であれば変更可である。

- METHOD :(例：-2)
タスクの状態を表すフラグ。0 は全てのユーザに承認された状態，その他は承認待ちであることを示す。
- NUM :(例：3)
タスクに関連のあるユーザの数。整数で記述する。
- NAME :(例：ham)
タスクの実行者の ID。複数人いる場合には次の STATE とともに繰り返し定義する。
- STATE :(例：-1)
タスクの受理状態を表すフラグ。-1 は未受理，0 は受理，1 は拒否を示す。
- START
- END
- PRIORITY
- TIMESTAMP
- TODO
- WORKLOAD
- ACCEPT
- SYNC
- PUB
- GROUP
- FROM
- MEMBER
- MEMO
- PROGRESS

その他のデータ構造

タスク情報の他にサーバが管理するデータとして以下のものが挙げられる。

- ユーザリスト
ユーザリストには本システムの登録ユーザの情報が格納されている。ユーザリストのデータ構造は次の通りである。

- ID : (例 : i2k)
システム上のアカウント . 英数半角文字で記述する .
- PASSWORD : (例 : * * * *)
システムの認証を受けるためのパスワード . 携帯端末で利用しやすいよう半角数字
4 桁で記述する .
- E-MAIL : (例 : ngrid@docomo.ne.jp)
携帯端末に付与された E メールアドレス . 管理者からの情報配信等に用いる .
- 認証ユーザリスト
認証ユーザリストはそれぞれのユーザが保持し , ユーザ間認証が行われた場合にエン
トリーが追加される . データ構造は次の通りである .
 - ID : (例 : ham)
相手のユーザ名 .
 - TIMESTAMP