

衛星開発における失敗防止のための電子メール処理システム

Email Processing System for Fault Prevention in Satellite Development

武田 光平[†] 加藤 義清[‡] 堀 浩一[‡]
Kohei Takeda Yoshikiyo Kato Koichi Hori

[†] 東京大学大学院 工学系研究科

Graduate School of Engineering, University of Tokyo

[‡] 東京大学 先端科学技術研究センター

Research Center for Advanced Science and Technology, University of Tokyo

1 はじめに

科学技術が発展するに従い、各種工業製品の開発過程は複雑さを増している。そのためにひとつのプロジェクトを一人の人間で全てまかなうことは基本的に不可能になり、複数の人間が協調して作業することを求められている。

複数の人間がひとつのプロジェクトにかかわるということは、必然的に情報共有と意思決定過程の複雑化をもたらす。開発過程においては設計パラメータや方針を開発者間で共有することが不可欠であるが、これらの情報は設計が進むにつれてしばしば変更される。これらの変更の伝達が十分に行われないと、プロジェクト自体の失敗につながることもある。例として、NASA の火星探査衛星である、Mars Climate Orbiter(MCO) があげられる。MCO は、1999 年 9 月に火星周回軌道への投入に失敗して消息を絶ったが、その原因は、航法を担当する 2 つのチーム内で「キロメートル」と「マイル」という異なる長さ単位系が混在して用いられており、航法パラメータに齟齬が生じたためであるとされている。

また、開発の過程においてなされた決定がどのような理由に基づいて行われたのか(設計意図, Design Rationale)という情報は、運用での不具合発覚時や次回以降の設計において特に重要になるが、定期ミーティングの資料や最終的な報告書には記載されないことが多い。この点で知識の再利用に関しても問題を抱えていると思われる。

これらの問題を総合的に解決するための枠組みとして、筆者らは、IDIMS(Integrated Design Information Management System)を提案している。IDIMS では、ドキュメント、メール、さらに設計ドメインの背景知識(オントロジー)を統合的に扱うことで、プロジェクトの失敗防止と知識の再利用を試みている(図 1)。具体的には、設計において特に重要となる問題(issue)と、それに対する決定(decision)というコンテキストに対し、設計ドキュメントやメールの対応する部分にタグを付加した上でデータベースに登録しておく。この issue と decision は未解決の問題を発見するのに用いられるほか、設計意図を明確化・保存するためにも使用される。さらに、将来的にはオントロジーを利用した推論を併用することにより、起こりうる不具合の可能性提示、すでに起こった不具合の原因候補提示などを行うことが可能になると期待される。

本稿では、この IDIMS プロジェクトの一環として、実際に宇宙科学研究所で開発中の小型衛星「INDEX」の開発者用メーリングリストに流れたメールに対しタグ付け・分析を行い、その結果を基に

- プロジェクトの失敗を未然に防止
- 次回の設計や運用に活かすことのできる知識を蓄積する

という観点からシステムを提案し、実験を行った結果を紹介する。

武田 光平 東京大学大学院工学系研究科
航空宇宙工学専攻知能工学研究室
東京都目黒区駒場 4-6-1
Tel:03-5452-5289 Fax:03-5452-5289
e-mail: taketa@ai.rcast.u-tokyo.ac.jp

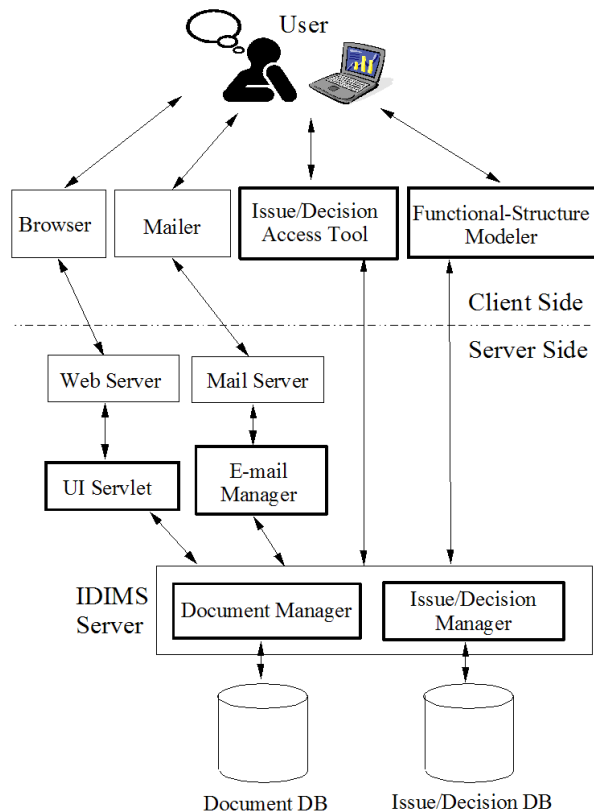


図 1: IDIMS 全体のシステム図
太字の四角が本研究の時点で実装されているモジュールを示す

2 メーリングリストによる共同作業支援にかかわる課題

人間の共同作業をコンピュータによって支援する研究は、CSCW(Computer Supported Cooperative Work)と呼ばれる一つの領域をなしており、その目的で作られたソフトウェアを一般にグループウェアと呼ぶ。一般に共同作業は時間と空間によって3つに分類される。すなわち、遠隔分散-非同期型、遠隔分散-リアルタイム型、対面-リアルタイム型である。本稿で述べるようなメーリングリストを用いた共同作業は、このうちの遠隔分散-非同期型にあたる。

メーリングリストはそもそも同報性を目的として構築されているため、情報を容易に複数の人間に通知できるという点では有用である。しかし、メールを本格的な共同作業のツールとして用いようとするれば次のような点が問題になる。

2.1 メール間の関連性のあいまいさ

一般に、意思決定やそれに伴う討論には、なぜその決定が行われたのかという理由(設計意図, Design Rationale)と、話題の全体像の把握が重要である。メーリングリストにおいて、これは問題を提起するメールと、それに応答する複数のメールを順序だてて読むことによって実現される。ところが、メール同士の関連性を示す情報は、ヘッダ部に記載された In-Reply-To もしくは、References 要素で規定される応答関係にしか存在しない。また、これらの情報の具体的解釈についてはメールクライアントソフトに任されているため、場合によってはこれらのヘッダが正しく付加されない場合もある。メールの応答関係が失われた場合、議論の流れを把握するために本文を読んで頭の中で組み立てなおすという余分な作業を強いられることになる。

この問題を解決するため、メールの書き手があらかじめ構造つきの内容を記述する研究がなされている。例として、The Coordinator[1]を代表とする構造化電子メールがあげられる。また、電子メールに限らず議論を構造化し、設計意図を明確化する研究として IBIS[2](Issue Based Information System)がある。IBIS では議論の論点を Issue (問題)、Issue の解候補を Position、Position の論拠を Argument として記録する。IBIS を実際のソフトウェア開発に用いた例として、gIBIS[3]がある。また、リアルタイム型の共同作業に適用した例として、rIBIS[4]がある。

IBIS と同様の研究として、QOC 言語 [5]があり、こちらは Issue、Position、Argument の代わりに Question、Option、Criteria を使用する。

一方、構造を書き手が入力するのではなく、コンピュータに自動的に抽出させようとする試みもある。村越らは、自然言語処理によりメーリングリストの討議構造を自動構築する研究 [6]を行い、一定の成果をあげている。また、ニュースリーダへの適用例として、総務省通信総合研究所の”HISHO”[7]がある。

2.2 知識の再利用の難しさ

もう一つの問題として、メールによるやりとりは大量かつ一過性であるため、知識として再利用することが難しいことがあげられる。従来メール資産の再利用法としては、MHonArc などのユーティリティを用いて WEB アーカイブ化し、メール単位で全文

検索することが行われていた．しかし，メール一通一通に含まれる情報は決して多くなく，前述のように応答メールを含めて一つの話題を構成するものである．そのため単一のメールに対するキーワード検索という従来の方法では，検索結果のノイズが多すぎたり，キーワード選択にノウハウが必要であるなどの不都合が生じる．

これに対し，前述の HISHO では，USENET ニュースに対し，単一の記事ではなく一連の記事で構成される「スレッド」単位で類似度を測り，類似する過去の話題を提示する試みを行っている．

3 提案するシステム

前節の問題点を踏まえ，次のような機能を備えるシステムを構築した (図 2) ．

現在本システムの備える機能は，以下の通りである．

- タグ付けされたメールを処理し，レポジトリ (データベース) に登録する．登録データ形式としては，XML を採用した．
- タグ同士のリンク関係をツリー上に可視化する． (図 3)
- WEB 上から，登録したメールを参照することができる． (図 4)
- タグ同士のリンク関係から，未解決の issue を自動的に検索し，提示することができる．
- issue から decision にいたる一連の議論構造 (スレッド) を単位として，検索を行うことができる．

3.1 タグ付けの文法

本研究では，IDIMS の issue，decision という構造をメールに対して導入した．これらのタグは，IBIS における Issue，Position，Argument を簡略化することで入力付加の軽減を図ったものである．また，メールに含まれる issue とは直接関係しないコメントや知識を拾えるように，notice というコンテキストもあわせて導入した．issue，decision，notice のコンテキストを表すタグで囲まれる内部には，それが他のどのコンテキストを受けて書かれたものかを示す，“inreplyto”タグを付加的に記述することができる．タグは，ちょうど $\text{\TeX}$ のコマンドのように，`\issue{` のような形

で記述する．実際にタグ付けされたメールの例を (表 1) に示す．

表 1: タグ付けされたメールの例

```
\issue{
\inreplyto{tagrel:0}
> > 極端に大きなバイアスがあるのであればハードウェアで調整すべきですが，
> > ある程度以上になるとハード的に押さえ込むのは非常に難しくなるため，
> > PCDYN のソフトで A/D ボードの値を電圧値に変換する係数を調整した方がよいと
> > 思います．
> 了解しました．
> 今日は難しいので，明日やります．
> この調整は AD ボードの個体差が大きいですよね？
> A さんをお願いするのは難しいかもしれない．
> # しかし，バージョンアップは A さん一人にしたいのだが...
この問題は，PCDYN 特有の問題ですか？ FM のハードでもおきる問題ですか？
FM の IOP，計表，RW の間のノイズのおおきさによって，ソフトで改修することも
あるのでしょうか？
\}
```

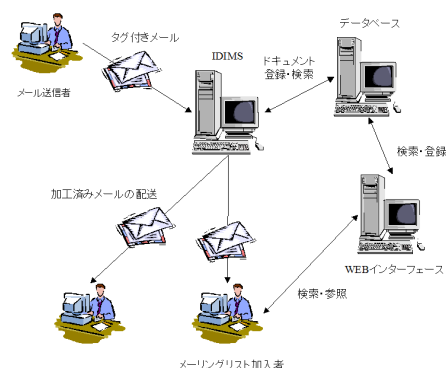


図 2: システム概観

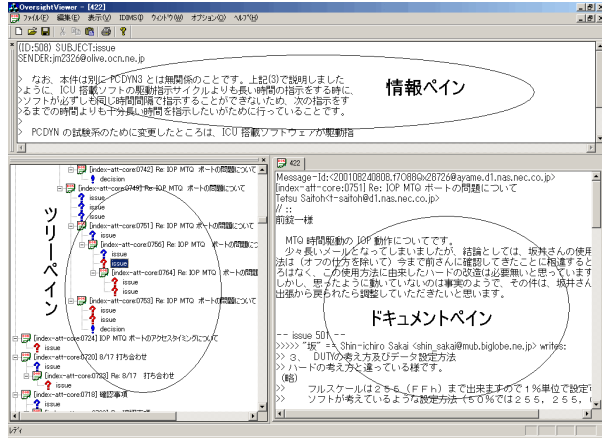


図 3: タグ構造の視覚化ツール

3.2 スレッド検索

前節の最後で述べたように、一通に含まれる内容が断片的である電子メールにおいては、メッセージ単位の検索ではなく、複数のメールで構成される話題単位での検索が重要である。幸い本研究では、タグ付けによりどのメールがどの話題に属するのかを特別な処理なしに知ることができる。そこで、これを利用して関連スレッドの検索機能を実装した。

3.2.1 スレッドの定義

通常メーリングリストにおけるスレッドとは、メールヘッダの In-Reply-To もしくは References 要素によって結ばれる一連のメールのことを指すが、本稿でのスレッドとは、inreplyto タグによってリンクを張られた一連のメール内文章ブロックの集合であるとする。よって、一通のメールの中に複数のタグ付けされたブロックが存在した場合、そのメールが複数のスレッドに属することもあり得る。

3.2.2 方法

メール登録の際にあらかじめ形態素解析エンジン「ChaSen」によってキーワードを抽出しておき、それを基に一連の議論を構造化した「スレッド」単位でのスコア計算を $tf \cdot idf$ で行った。本研究では、本来文書単位で定義される $tf \cdot idf$ を拡張し、スレッド単位

Subject: IDIMS-MAILの変更について

From: taketa@aircast-u-tokyo.ac.jp
To: idims-spokes@aircast-u-tokyo.ac.jp
Date: 2001-11-1 14:51:40
Message-ID: 200111010547.0AA15286@annie.aircast-u-tokyo.ac.jp
References:

M2武田です。

Notice from taketa@aircast-u-tokyo.ac.jp
Keywords: 変更, IDIMS, メール
Description:
idims-spokes@aircast-u-tokyo.ac.jp
で稼動中のIDIMSメールサブシステムの変更をお知らせします。
1. コンテキストの始まり・終わりのメッセージを簡素にした。
2. 詳細なreference情報などはメールには表示せず、WEB側でみる仕様とする。
3. 「このメールはIDIMSによって処理されています」云々のメッセージは、メール末尾、返信用テンプレートの直前に移動。
4. 返信用テンプレートを添付ファイルにするのをやめ、本文の末尾につけかわる形にした。これで、元のメールに返信し、前半を削除するだけで返信メッセージの雛型ができる。
References:

Issue raised by taketa@aircast-u-tokyo.ac.jp
Keywords: 変更, IDIMS, 形式, メール, タグ
Description:
ところで、コンテキストタグの形式について、現在のように

図 4: WEB インターフェース

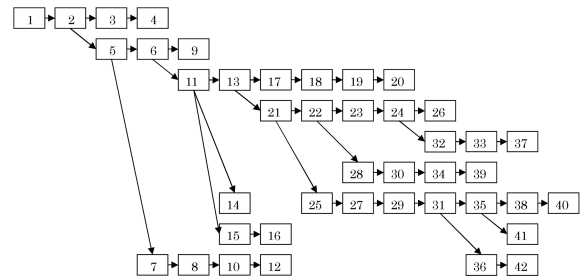


図 5: メール単位での議論構造例

での $tf \cdot idf$ として、以下のように定義した。

$$tf = f/m$$

$$idf = 1 + \log(N_t/n_t)$$

$$tf \cdot idf = tf \times idf$$

ここで、

f はあるスレッド内での検索キーワード出現頻度

m はスレッドに含まれるタグ数

N_t は全スレッド数

n_t はキーワードを含むスレッド数

を表す。

これを用いて、スレッド t' に対するスレッド t の検

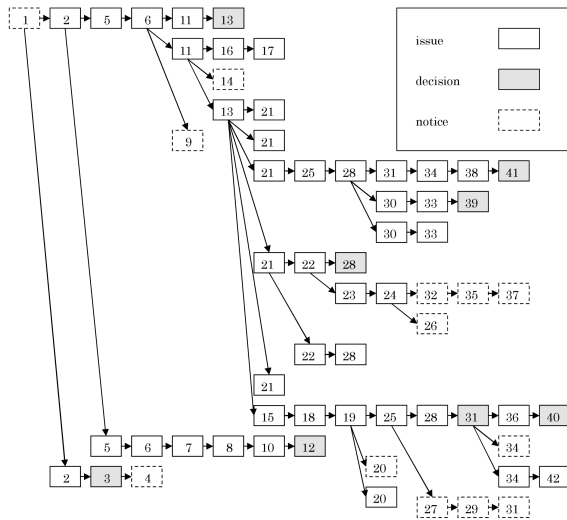


図 6: タグ付けによる実際の議論構造例

索スコアを，以下のように定義した．

$$score(t', t) = \sum_{k \in K(t')} tf \cdot idf(k, t)$$

t' : 類似スレッド検索対象となるスレッド

T : 全てのスレッドを含む集合

$K(t')$: スレッド t' に含まれるキーワード集合

4 実際のメーリングリストログへの適用と考察

上で述べたシステムを用い，実際に宇宙科学研究所で開発中の小型衛星，INDEX の開発者メーリングリストの 2001/4/24 ~ 2001/10/10 までの 679 通のメールにタグ付けし，解析を行った．

4.1 未解決のまま放置されている問題の提示

本システムを使用することで，解析したメールのなかから対応する decision のない issue を 84 個見つけることができた．その内訳は以下の通りである．

タイプ	項目数
issue と decision が直接つながりのないメールに存在	11
issue が直接つながりのないメールで継続	2
暗黙に解決したとみなせる	9
返答なし	56
不明	6
合計	84

「返答なし」については，issue に対する decision が対象としたメール群に存在しなかったものである．「不明」とは，issue の記述が不明確であり，筆者にとって何についての問題なのかが判然としなかったものである．「返答なし」と「不明」の合計値が，未解決のまま結論に至らない問題数であると考えられる．

解析したメール総数は 679 通であるため，約 11 通に 1 つ，対応する decision のない issue が存在したことになる．これは，衛星開発という不具合に対して厳しい条件を課せられるプロジェクトでの結果としては，見過ごすことのできない数値である．

ここで，実際に行われたメーリングリストでの議論の特徴と問題点を示すため，未解決 issue を含むスレッドの例を巻末に付した．(図 5) は多くのメール閲覧ソフトで実装されている，In-Reply-To，References 等のメール間返信関係を用いた構造表示である．ノードに付された番号は，メールが出された順番を表している．それに対し，図 E.2 は，issue と decision によるタグ付けを行い，実際の議論構造を視覚化したものである．ノードに付された番号は，そのタグが含まれるメールの順序番号である．この例を見てわかるように，メールを単位として考えた場合の構造と，実際に行われている議論の構造には食い違いがある．特に，一つのメール内に複数の issue が存在している場合に，一見 (図 5) 側では応答が行われているように見えても，実はいくつかの issue が未応答のまま残される例が目立った．例として 21 番のメールでは 5 つの問題が提起されているが，そのうち 2 つにしか応答がない．これは，構造化されていないメールで議論を行ううえでの，人間の認知的な限界を示しているものと思われる．この問題を解決するためには，議論の全体像を把握するための視覚化サポートと，未解決問題の自動警告が有効だろう．

4.2 知識の不十分な活用について

上記の問題以外に、以前結論の出た内容を繰り返し質問したり、一度現れた情報が、その時点では注意を払われなかったものの、後に不具合の原因となっている事例も確認された。後者の例のあらましを述べると、以下ようになる。

1. あるセンサのデータ受け渡しが正常に動作しているか確かめるため、センサの電圧出力 0-3V を 10bit のデジタル入力に変換して計測する試験を行った。このスケーリングを行ったという事実は、その日の試験報告メールに記載されている。
2. その際、現場での専門家のアドバイスにより、電圧の出力範囲に 10% のマージンを持たせ、10bit(0-1023) ではなく、0-926 程度の値にマップピングして計測を行った。
3. その後、日をあけて再度試験を行ったが、10% のマージンをとったスケーリングを行っていることを忘れ、その分のズレを誤差として認識。周辺機器の調査依頼を行った。
4. 結局原因がわからず、10% の誤差を正常範囲にあるものとしてかたづけた。
5. 後日、試験結果の誤差がバイアス的なものか、スケールのなものかという質問が他の開発者からなされる。
6. そこで試験者が、最初にマージンをとっていたことを思い出し、解決。

この例では、最初のメールの段階ではスケーリングを施したという事実が issue と認識されず、単なる報告として現れていることに注目すべきである。つまり、前述の議論の視覚化サポートや未解決の issue を警告するだけでは解決できない問題である。

4.3 関連スレッド検索

そこで、過去の知識を効率的に検索し、設計知識を含む一連の構造単位で結果を提示できるよう、スレッド検索の実験を行った。評価方法としては、前節の「以前の決定を忘れて質問を繰り返す場合」及び「不具合の兆候が、すでに以前のメール内に存在して

いる」案件（計 3 件確認した）についてスレッド検索と Namazu によるメール単位での全文検索（スコアリング方式は *tfidf* を指定）を行い、原因となるスレッドが検索結果のどの順位に現れるかを比較した。Namazu の検索についてはキーワードが必要なため、筆者がメール内容を良く表していると判断したキーワードを 1 つ、もしくは 2 つ組み合わせて指定した。スレッド検索については、問題が認識され、問いが発せられたメールの内容をキーとした。

例	スレッド検索順位 Namazu 検索順位
1	19 位 ”トルカ”で 2 位。 ”MTQ”でヒットせず
2	2 位 ”FOG”で 10 位。 ”FOG+極性”で 6 位
3	18 位 ”GAS”でランク外。 ”GAS+スケール”で 7 位

一見 Namazu の検索結果と変わらず、むしろ Namazu の方が良い結果を出している部分もあるが、実際には、Namazu の検索結果はキーワードの与え方に大きく依存する。たとえば、例 1 では Namazu の検索順位は 2 位だったが、「トルカ」という単語は、全メール中でほとんど出現しない単語である。同義の単語である「MTQ」の場合は頻繁に出現するが、この場合は Namazu ではヒット数が多すぎ、20 位以内に対象メールが出現しなかった。例 3 に対して「GAS」を単独で与えた場合も同様である。プロジェクトに精通している人が検索を行う際には Namazu でも事足りるだろうが、たとえばこの資料を後で手に入れて別プロジェクトで使おうとした場合、同じく適切なキーワードを用いて検索ができるかどうかは不明である。これに対して、スレッド検索の場合はコンスタントに 20 位以内に検索結果が現れている。この結果をもとに、今後は語の共起関係や、シソーラス等を用いて検索精度を上げることが課題になるだろう。

5 成果

メールログの解析により、実際の衛星開発の現場に置いても、問題が未解決のまま放置されることがや、すでに結論の出た議論が繰り返されることがあることを示した。

このうち前者については、従来のメール単位での構

造表示と実際の議論構造は異なり，それが問題を未解決のまま放置することにつながっていることを示した．すなわち，本研究で用いた issue，decision リンクの視覚化や，未解決問題の自動検索と提示サポートによって実際の議論構造把握を促進すれば，これらの見落としも減少すると思われる．

また，後者については，メール単位ではなく一連の議論を構造化した「スレッド」を単位として検索をおこなうことにより，蓄積されたメールに存在する知識を特別なノウハウを必要とせず，より人に分かりやすい形で提供できることを示した．

6 今後の課題

本論文の段階では，製作したシステムを現実の設計現場で運用するには至っていない．今後は実際のプロジェクトに適用し，本当に未解決の問題が減るのか，知識が蓄積・活用されるか，また現行のメールシステムに比べてどの程度の認知的な付加をユーザに課すことになるのかを検討する必要がある．

また，本研究は IDIMS プロジェクトの一環であり，将来的にはメールだけでなく一般の文書や，オントロジーとともに総合的な設計情報システムとして機能することが期待される．この統合という観点からも，今後研究を行う必要がある．

謝辞

INDEX プロジェクトのメールログの利用をご快諾頂いた宇宙科学研究所の齋藤宏文教授に感謝致します．また，メールログの提供にご協力頂き，研究にコメントを下された宇宙開発事業団の福島洋介氏に感謝致します．

参考文献

- [1] Winograd, T. and Flores, F.: Understanding Computers and Cognition, Addison-Wesley Publishing Company, Inc, 1986
- [2] H. Rittel and M. Webber. Dilemmas in a general theory of planning. *Policy Sciences*, Vol.4,, 1973.
- [3] Jeff Conklin and Michael L. Begeman, gIBIS:A Hypertext Tool for Exploratory Policy Discussion, CSCW'88,ACM,September 1988,pp.140-152.
- [4] Rein, G L and Ellis, C A. "rIBIS: a real-time group hypertext system". *International Journal of Man-Machine Studies* (1991) Vol 34. pp349-367.
- [5] Shum, S. B. and Hammond N. Argumentation-based design rationale - what use at what cost. *International Journal of Human-Computer Studies*, 40(4), pp.603-652, 1994.
- [6] 村越広享，島津明，落水浩一郎，メーリングリストを利用した共同作業における討議構造の自動構築法，コンピュータソフトウェア，Vol.18 No.3 pp.19-23, 2001
- [7] 知的ニュースリーダー「HISHO」
<http://www.tis.co.jp/research/ipa-hisho/>