

CS と CW の統合に関する考察 ～ CS から CW へ、そしてまた CS へ～

垂水浩幸

tarumi@acm.org

京都大学大学院工学研究科 情報工学専攻

これまでの CSCW の研究を振り返ると、当初の技術中心から、最近では社会学系の研究が非常に盛んになってきている。しかしながら、両者の研究の連携がうまくいっているとは言えない。また、大規模なグループウェアはその効果の説明が困難であるため現場で導入しにくいという問題がある。本稿では、非リアルタイム型のグループウェアの開発ライフサイクルにおいて、現場の観察成果を設計に生かす過程について議論する。このようなライフサイクルは、グループウェアの有効性を実証し、グループウェアの現場導入に対しても有効なはずである。

A New Approach to Integration of CS and CW — CS to CW, and back to CS —

Hiroyuki Tarumi

Department of Information Science, Kyoto University

Looking back at the history of CSCW research, we can find that the number of research papers with social science approaches are increasing in contrast to the early days when most of the papers were with technical approach. This paper proposes a development lifecycle for groupware, which can utilize results of social studies at the early stages of lifecycle. This lifecycle is also helpful to show the effectiveness of the groupware, and to encourage people to introduce the groupware into the fields.

1 はじめに

1986年に最初のグループウェアの国際会議(CS CW'86)が開始されてから、10年以上が経過した。CSCW (Computer Supported Cooperative Work) はCS (Computer Support) を研究する計算

機科学者、技術者とCW (Cooperative Work) を研究する社会学者、心理学者とが協力して、新しい研究分野を確立しようとするものであった。当初CS主体であったCSCW会議も90年代になって徐々にCW側の参加が増え、最近の様子を見るとほぼ

対等の勢力になったように思われる。

筆者は 97 年度に開催された二つの CSCW 関係の会議である ECSCW'97 と Group'97 に参加した。ECSCW の開催された英国のランカスター大学は計算機学者と社会学者の協力が最もうまくいっている研究機関の一つである。またヨーロッパでは全体的に社会学者の CSCW 分野への参入が盛んである。こういった背景もあって、ECSCW'97 では社会学的アプローチの論文が多く、技術的論文も何らかの形で社会学的アプローチを意識したシステム評価を行なっているもののみが採録されていた。また採録された論文はほとんどがヨーロッパ内のものであった。

Group'97 では論文の受付窓口がテクノロジー側とソーシャル側との二つに別れており、それぞれの分野から均等に論文を採録するという運営方針であった。ここで特筆すべき点は、この会議が米国で（しかもヨーロッパから遠いアリゾナで）開催されたにもかかわらず、ヨーロッパの論文の方が米国の論文より多かったということである。これは社会学的アプローチの論文の数が増えていることと無縁ではないと思われる。

このような状況を観察すると、CSCW 研究分野の学際化は 10 年という時間はかかったものうまくいったように見える。だが本当にそうであろうか。

率直に言って、最近の CSCW 関係の学会で発表される論文に面白いものが少ない。もちろんこれは全くの個人的感想であるが、多くの研究者、特に技術系の研究者は同じような感想を持っているのではないだろうか。もしも CSCW 分野の学際化が成功していれば、この時期は両者のコラボレーションによって新しい種類のエキサイティングな論文が出てきていてもおかしくないはずである。

以下に、この沈滞現象の原因を技術系と社会学系の両者について検討する。もちろんこれは技術系の研究者としての立場での検討である。

まず技術系であるが、1990 年前後はリアルタイム系、非リアルタイム系を問わず会議システム、メディアスペース、仮想空間、知的電子メール、議論支援システム、ワークフローシステム等、新しい概念が次々に現われ、それらが実際のシステムとして稼働していく時代で非常にエキサイティングであった。しかし最近の論文は既存システムの実用評価や WWW を応用したシステムの焼き直し等の論文が多い。これらの仕事はそれなりに有益だが、エキサイティングなものとはいえない。こうなってしまった原因としては

- 主な協調作業のパターンに対して、技術的に支援できることはほぼ出尽くした。
- 特にリアルタイム系の通信では技術に対してコストがかかることが多く、商業的に成立しにくい。
- 技術を駆使しても、それが実際に役に立つのか、ユーザが使いこなせるのか、ユーザがそれを受け入れるのかといった議論が CSCW の分野では特に厳しく、論文を書きにくくなった。

などが考えられる。

一方社会学系であるが、90 年前後の論文では、グループウェアを実際に役立てようとしても様々な理由でユーザがそれを受け入れない [1, 2] ということが我々技術者にとってショックであり、非常に参考になった。しかし最近の状況は

- コンピュータ支援のグループ作業を社会学的に分析した論文は多いが、その分析結果はおよそ予想の範囲内で、情報にはなっても驚きを与えるようなものではない。
- 分析が分析に終わってしまっており、それをグループウェアの設計にどう役立てるかの議論が弱いのでは技術者にとって面白くない。
- 分析は特定のアプリケーション分野に対して行なわれることが多いので、他のアプリケーション分野には役立たないこともある。

という問題を感じざるを得ない。すなわち、CSCW は社会学者に新たな研究対象を提供しただけであって、社会学者と技術者との創造的な共同研究にはなっていないのではないかとと思われる。

以下では、現場観察に代表される社会学的アプローチの成果を、グループウェアの設計・開発に如何に還元すべきであるかを議論し、グループウェア開発のライフサイクルモデルを提案する。

2 本研究の対象

本論に入る前に本研究で対象としているグループウェアの範囲を明確にしておく。本研究では、今後重要になると思われる下記の条件を備えたグループウェアを対象としている。

1. 開放的であること

複数の事業部を持つ大企業、複数企業の連携、自然発生的な組織（同好会、ボランティアなど）、バーチャル企業など広い分野での利用を想定し、開放的であることを前提条件と

する。これは、具体的には以下の条件を満たすことを意味する。

- 各ユーザが複数のワークグループに同時に所属することを前提とする。このためには、複数のグループから依頼される仕事の同時処理に対する支援 [3] や、個人作業とグループ作業の同時並行支援 [5] が必要になる。
- イン트라ネット、エクストラネット、インターネット等の言葉で表現される広域ネットワークをベースとする。
- 組織間のセキュリティを配慮する。複数の組織の連携作業では、ある種の情報をファイアウォールで保護する必要がある。
- マネージメントが分散的である。複数組織の連携作業や自然発生的組織では、それぞれの組織の長あるいは各構成員のマネージメント方針の違いを認めなければならない。
- 文化の違いを容認する。組織毎に用いる用語の違い [4]、年中行事などのスケジュール、仕事の優先順位の違い、管理の形態の違いなどに対応できる。

2. Scalability を持つこと

システムには随時新たな要素を追加したり、あるいは要素を削除したりできなければならない。また、規模の大小にかかわらず動作しなければならない。

3. 発展的であること

そのときどきの例外的なイベントや事情にダイナミックに適応できるばかりでなく、プロセスの定義の変更などの発展的な変更にも随時対応できる。

これらの条件を満たすグループウェアの構成方法としては、その分散性、自律性を考慮するとマルチエージェントを基礎とするものが最適であると考えられる。

さらに、本研究では主として非同期通信に基づく協調作業支援 (非リアルタイム型グループウェア) を対象とする。これは、リアルタイム型のグループウェアの場合、上記に述べたような特徴を採り入れても新しい研究課題があまり発生しないと思われるためである。

3 グループウェア導入上の問題

3.1 問題点

筆者はこれまでワークフローシステム、グループスケジュール管理システムなどの非同期系の通信を用いたオフィス応用を中心にグループウェアを開発し、現場への導入を試みてきた。しかし設計・導入においては次のような困難に直面した。

効果が予測できない

実際にそのグループウェアを現場で利用した場合、どれだけの効果があるのか定量的に予測するのが困難である。現場の管理者は、実際的な効果が期待できることの説得力ある説明を必要とする。直観で定性的な効果を期待して導入してくれる管理者はごく少数である。したがって、グループウェア設計者は現場の導入責任者に対してそれを説明する必要がある。

ところが、それは容易ではない。まず、グループウェアの効果というもの定量化すること自体困難である。なぜならば、特に非同期系グループウェアの効果は人間同士のコミュニケーションがうまくいかないことによるロスを低減化するという点に主に現れることが多く、主たる作業 (例えば文書作成) 自体を効率化するものではないからである。

次に、グループ作業の現場の実態がそれぞれの現場で異っているため、各現場での実態に即した説明が求められる。これも困難な課題である。特に上記のようにロスを低減化することがグループウェアの目的であるとするならば、現場における無駄の現状を調査する必要がある。そのようなデータは現場の管理者は出したがらないし、またロスがあることを意識的あるいは無意識的に認めようとはしない。

実用評価が困難

上記のようにグループウェアの効果を定量的・一般的に説明することが困難であるため、グループウェアの効果を説明する方法として、実験的な現場での実用評価を行ってそれを証拠として効果を説明することが考えられる。これは一つの有力な方法であるが、例えばワークフローシステムの場合、一定以上の人数の現場の人に試用してもらわなければならない。しかし、現場で現に行われている作業を実験的なシステムで置き換え、さらに一定以上の人数の人を説得して試用してもらうことは容易ではない。

また、大規模で開放的なグループウェアを考えた場合、小規模の実験だけでは大規模に利用した場合の効果が約束できない。さらに、たとえば防災システムのような特殊な状況での使用を想定したシステムの場合、試験運用自体不可能なこともある。

例外的な成功例としては POLITeam がある。POLITeam では、ドイツ政府のボンからベルリンへの首都機能移転に合わせて試験運用が行なわれたが、その際、以下のようなプロセスによる、利用者参加型の設計が行なわれた [6]。

1. システム導入の準備: ユーザインタビューによる要求分析、システムの調整
2. システムの導入: トレーニング、インストール、現場訪問
3. 現場実用による評価: 現場訪問、ユーザーワークショップ、インタビュー
4. システム設計: 再設計と新機能の設計のためのデザインワークショップ
5. ステップ 2 に戻る

この例では、ユーザー参加のワークショップを繰り返し行なって良い結果を得ている。成功原因としては、首都機能の段階的移転による大規模分散作業という状況において、トップダウンの強い意志によるところが大きいと思われるが、一般の例でこのようにうまくいくとは期待できない。

人間の振舞が予測できない

Grudin [2] も指摘しているが、グループウェア設計者は、現場の利用者が日常どのように振舞っているか、またグループウェアを導入した場合にどのように振舞うかを予想できない。したがってグループウェアの効果を想定した設計ができない。

まとめ

以上まとめると、グループウェアの効果を説明するためには、(1) 現場の理解、(2) 可能なら現場での試験運用、(3) 大規模な運用を行った場合の予測という三段階が必要ことがわかる。引き続き、これらの各段階について考察する。

3.2 現場の理解

現場の理解という点については、二つのアプローチが考えられる。

1. エスノグラフィックアプローチ:
モデルなどの前提を持たずに現場を客観的に観察した結果をもとに考察する。

2. モデルベースのアプローチ:

ユーザの振舞いについてあるモデルを当てはめ、そのモデルに基づいて説明する。

前者はグループウェアの要求分析段階で、後者はグループウェアの設計段階で有用であると考えている。

3.3 現場での試験運用

現場での試験運用についても二つのアプローチが考えられる。

1. 個人別運用

グループウェアを個人が利用した場合、その個人の振舞いが設計の想定通りであるかどうかを検証するための運用。例えば、「ユーザは To-Do リストに記載された仕事のうち細かい仕事から先に手をつける」というモデルに基づいて設計されたシステムがあるとするならば、実際にユーザがそのように振舞うかどうか検証する。これはグループで運用する必要はないので試験運用が容易にできる。

2. グループ運用

小規模のグループで実際に運用し、想定通りの効果やグループの振舞いが見られるかを検証する。

3.4 大規模運用の予測

大規模な試験運用は実際にはほとんど不可能である。そこで、上記の手順で得られたユーザの振舞いのモデルをもとにして、シミュレーション検証を行う必要がある。特に、スケーラビリティの観点からの評価が必要である。

次章では、本章の考察をもとにして、具体的にグループウェア開発手法を提案する。

4 本研究の提案するライフサイクル

図 1 は、本研究で提案するグループウェア開発のライフサイクルである。以下、順を追って説明する。

4.1 要求分析

グループウェアの目標、主な機能を設定する。この際、現状何が問題か、何を支援すべきかなどの決定には現場の客観的観察が必要であることから、エスノグラフィック・アプローチなどの観察手法が有効と予想できる。

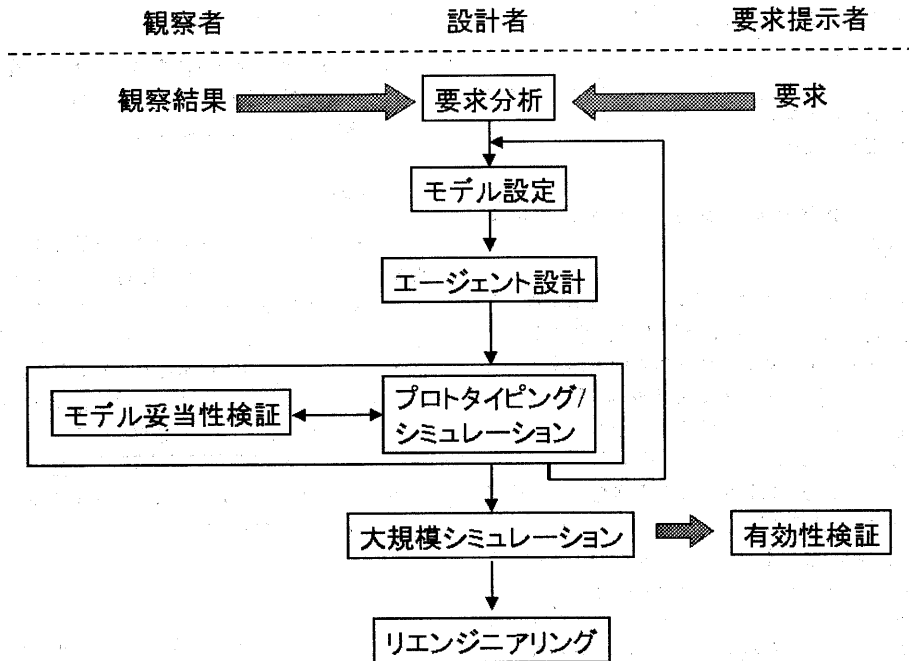


図 1: グループウェア開発のライフサイクル

ここで、要求提示者は現場のマネージャなど、グループウェア導入を検討中の人物を指す。これとは別に観察者の存在を明確にする。観察者にはコンサルタントのようなマネジメント技術の観点から現場の問題点を指摘する性格の人物と、社会科学的アプローチによって現場の状況をレポートする人物とが考えられる。(両者の間に明確な線引きはないが。)

4.2 モデル設定

支援する対象のグループの仕事やコミュニケーションをどんなモデルで表現するかを検討し、モデルを設定する。例えば、人と人とのコミュニケーションではどのような言語あるいはデータがやりとりされるのか、人の振舞いを記述する方法はあるか(例えば人間の振舞いを状態機械として模擬できるか)、外界から生じるイベントにはどのようなものがあり、それはいつ起こるのか、等である。

ここで、外界とはグループウェアの対象以外の人物や組織を意味する。外界から生じるイベントとは、例えば顧客からの注文、為替変動、競合他社の動向などがある。もちろん「顧客」もグループウェアの支援範囲に含まれている場合は顧客は外界にな

らない。

4.3 エージェント設計

モデル設定のフェーズで、グループ作業における各作業者のふるまい、他の作業者とのコミュニケーションが明らかになる。次のエージェント設計の段階では、作業者の機能を人間とエージェントに分離して分担を明確にする。その際、エージェントと人との間のインタラクションモデルも明確にする。他の作業者とのコミュニケーションの一部(あるいは全部)はエージェントを介して行う。

4.4 モデル妥当性検証

設定されたモデルとエージェントの設計を念頭において、現場の観察を再度行ない、あるいは過去の観察結果を振り返り、モデルの妥当性を検証し、モデルの各パラメータ(例えば仕事を行なう時期、コミュニケーションの頻度等)を規定する。また外界のイベントについてもパラメータを規定する。

モデルが不適当であればモデルの設定からやり直す。

作業者の振舞いは作業者毎に異なるので、全員に対して同じモデルが適用できるわけではない。どの

ようなモデルをどのような人達に適用できるかも議論すべきである。

もちろん、グループウェアの導入によって仕事の仕方が変更される場合、その部分についてはこの段階では検証できない。

4.5 プロトタイピング / シミュレーション

試作を行ない、小規模のグループで試用する。試用の結果、再度モデルの妥当性が検証される。グループウェアの導入によって仕事の仕方が変わった部分についてはここで初めてモデルが検証されるので、必要に応じてモデルの修正を行なわねばならない。

グループ全員の同時試用が困難な場合は、各作業員について個別に試用し、モデルの検証・修正を行なう。この際、外界や他の作業員については模擬プログラムによるシミュレーションで代行する。

ここまでのフェーズは、理想的には、モデルの妥当性が確認されるまで繰り返されるべきである。

4.6 大規模シミュレーション

大規模なグループ、複数の組織で連携して本グループウェアを利用した時、どのようなことが起きるか、どのような効果があるかをシミュレーションによって検証する。このシミュレーションは作業員の動作を模擬するプログラムとエージェントとを連携させることによって行なう。作業員の動作を模擬するプログラムもまたエージェントの形で作られることになるだろう。

大規模使用では、プロトタイピングのフェーズまでで想定した小規模グループ以外の作業員が新たに多数加わることになる。これらの新しい作業員に対して既存のモデルをそのまま当てはめて良いかどうかは十分に検討する必要がある。特に他組織でもキーマンについては独自に振舞いのモデルを再構築する必要がある。

シミュレーションの履歴を解析することによって、グループウェア導入効果を定量的に予測できる。これによって、グループウェアの有効性を示し、導入を促進することができる。

また、最低何人の利用者がグループウェアを積極的に利用すれば効果があるかということもわかるので、いわゆるクリティカルマス (Critical Mass) の予測なども行なえるのではないかと期待できる。

4.7 リエンジニアリング

導入後の仕事の内容や仕事のプロセスの変化、メンバーの変化等に対応してモデルの修正、エージェントの機能の修正等が必要になる。

5 おわりに

グループウェアの現場での導入を容易にしつつ、開発コストを下げることを目的とし、シミュレーションを核とした開発ライフサイクルについて提案した。このライフサイクルの上流部分では、社会的アプローチによる現場調査が有効になる。

今後は上流工程における手法の明確化が課題である。また、シミュレータの開発を中心に研究を進めて行く他、シミュレーションの結果を分析するのに有効なユーザインタフェースやツールについても研究を広げていく予定である。

参考文献

- [1] Grudin, J.: Why CSCW Applications Fail: Problems in the Design and Evaluation of Organizational Interfaces, *Proceedings of the ACM 1988 Conference on Computer Supported Cooperative Work*, ACM (1988), pp.85-93
- [2] Grudin, J.: Groupware and Social Dynamics: Eight Challenges for Developers, *Communications of the ACM*, vol.37, no.1 (1994), pp.92-105
- [3] Holt, A.: *Organized Activity and Its Support by Computer*, Kluwer Academic Publishers, Dordrecht, The Netherlands (1997)
- [4] Poltrock, S.E. and Engelbeck, G.: Requirements for a Virtual Collocation Environment, *Proceedings of ACM Group'97*, pp.61-70 (1997)
- [5] Ishiguro, Y, Tarumi, H., Asakura, T, Kusui, D, and Yoshifu, K: "An Agent Architecture for Personal and Group Work Support", *Proceedings of International Conference on Multi Agent Systems '96*, pp. 134-141 (1996)
- [6] <http://www.orgwis.gmd.de/projects/POLITeam/design/>