■S3 群（脳・知能・人間）- 8 編（コラボレーションシステム）

5 章 CSCW の応用

（執筆者：市村　哲）[2010 年 1 月 受領]

■概要■

遠隔地にいながらにして作業者に指示を与えることができる遠隔作業指示，及び，開発者の協調活動を支援する設計活動支援は，典型的な CSCW の応用分野である．5 章では，CSCW の応用として，遠隔作業指示と設計活動支援を取り上げ，関連する用語などについて解説する．またこれら CSCW 応用技術を概観し，関連文献を紹介する．なお，医療と教育への応用は別編に譲ることとする．

【本章の構成】

5-1 節では，機器操作または修理を行わなければならない作業者（または学習者）が，コミュニケーションメディアを介して物理的に離れた場所にいる指示者から遠隔指示を受けることができる環境を実現する技術について解説する．5-2 節では，多人数の開発者の協調活動と見なすことができるソフトウェア開発を，ソフトウェアの設計・開発プロセスにおいて支援する方法論またはツールについて解説する．

■S3 群 - 8 編 - 5 章

5-1 遠隔作業指示

（執筆者：葛岡英明）［2009 年 2 月 受領］

遠隔作業指示（remote instruction）とは，指示者（instructor）が学習者（learner）に対し，コミュニケーションメディアを介して遠隔的に作業の指示を行うことをいう．この場合，コンピュータ画面上の GUI 操作に関する作業指示もあり得るが，これは「アプリケーション共有〔1 章 1-5 参照〕」，「グループエディタ〔3 章 3-2 参照〕」に近いため，ここでは扱わない．本節では，物の組み立てや機器の操作・修理など，物理的な対象物に対する作業の遠隔指示について解説する．対話者どうしがより対等な立場にある場合には，協調的物理作業（collaborative physical task）といい，この場合，対話者は支援者（helper）と作業者（worker）と呼ばれることもある[11]．しかし，遠隔作業指示と協調的物理作業で議論されている内容には共通点が多いため，本節ではこれらを区別しないものとする．

5-1-1 実物体参照

複数の人が同じ対象物に対して注意を向ける過程は「共同注意（joint attention）」と呼ばれ，人間ロボットインタラクション（human-robot interaction）などの応用分野において，その支援の必要性が指摘されている[12]．共同注意の過程において特に重要なのは，対象物に向けられた視線や指さしを相互に認識することである．こうしたことは指示者と学習者が同じ空間にいる状況では容易であるが，双方が遠隔地に分散した状況では，音声や実画像通信を利用した技術的な支援が必要である．

実物体参照に対する技術的な支援は，単に音声・画像通信が提供されればよいわけではない．例えば，一般的なテレビ会議システムを利用して，対話者の環境内にある実物体をさし示そうとする場合を考えよう．このとき，仮に対象物を指さししても，この指さしはテレビ画面上に表示された対象物をさしているにすぎない．従って，そのジェスチャをテレビ画面越しに見ている対話者は，自分の環境内のどこを指さされているのか理解することができないのである[5]．特に遠隔作業指示において考慮すべき身体的行為は，指さし以外に，対象物の動きや操作方法を示す手振り，身体的な志向（胴体，頭部，視線などの向き）などである．

1975 年に発表された Chapanis の研究[2]では，遠隔作業指示や遠隔共同問題解決において，音声・画像通信，音声通信のみ，テレタイプライターによる通信などを比較した．彼の実験では遠隔的な指さしは支援されておらず，画像通信による顕著な効果は見られなかった．

これに対して，1990 年前後には実物体に対する指さしや手振りを可能にする研究がみられるようになった．例えば，指示対象物と教示者の手振りを合成した映像を学習者に見せられるシステム[7,6]，遠隔操作型レーザポインタを利用して，教示者が指示対象物を直接ポインティングできるシステム[9]などがある．今後は，携帯電話のような小型デバイスにカメラとプロジェクタが搭載されることが期待されており，携帯型の端末が遠隔作業指示に利用できるようになると期待できる．

5-1-2 予期

人々がコミュニケーションをするときに示される様々な発話や身体的な行為を観察することによって，人は，対話者がこれから何をしようとしているのかを予期することができる[1]．

例えば，学習者は指示者の身体的な志向の変化を観察することによって，その対話者が次にどこへ行こうとしているのか，どの対象物について説明しようとしているのかということを予測することができる．予期が可能であるということ（projectability, predictability）は，遠隔作業指示における実物体参照を容易にする上で重要である．

この予期の問題に対しては，教示者の頭部志向の変化を教示者の代理ロボットの頭部の志向の変化としてリアルタイムに提示することのできるシステム[8)]や，遠隔操作型のレーザポインタにおいて，レーザポインタの光路を常に観察可能とすることによって，教示者の志向の変化になるべく早く気づかせることができるシステム[10)]が提案されている．

5-1-3 評価手法

遠隔作業指示を支援するシステムの有効性を評価するためには，定量的な手法と定性的な手法の双方が組み合わせて用いられることが多い．定量的な手法は，正しく用いられれば客観性が高く，説得力もあるが，実際にどのようなコミュニケーションが行われているのかということを詳細に理解することが難しい．これに対して定性的な手法は，様々な発話や身体的な行為が作業指示においてどのような役割を果たしているのかということを詳細に理解することに役立つ．

定量的な手法では，作業指示に要した時間を計測し，比較する場合が多い．しかし，これだけではどのような理由によって差が生じたのかがわからない場合もあるし，タスクによっては時間的な差が生じないこともある．そこで，時間分析に加えてよく使われるのが発話数の分析である．これは，コモングラウンド（common ground）[3)]*形成のための対話的グラウンディング[4)]（conversational grounding）†という考え方に基づいている．教示者が学習者に対してある対象物を参照させる際には，様々な言語表現を駆使してグラウンディングしようとする．そこで，作業指示中に用いられるメッセージ長，話者交代の数，指示語の数などを数えることによって，コミュニケーションの特性を評価できると考えられている．

定性的な分析としては，エスノメソドロジー的な手法を用いることが多い[13)]．これは，作業指示の際の発話，視線の移動，その他の身体的な動作などを詳細に記述し，いかにして対話が成立しているのかということを分析する方法である．定性的な手法と定量的な手法は相容れないという主張もあるが，ユーザインタフェース（user interface）の分野では，これらを併用する傾向が強まっている．

■参考文献

1) P. Auer, “Projection in Interaction and Projection in Grammar,” Text, vol.25, pp.7-36, 2005.
2) A. Chapanis, “Interactive Human Communication,” Sci. Am., vol.232, pp.36-42, 1975.
3) H. Clark, “Using language,” Cambridge University Press, 1996.
4) S. Fussell, R. Kraut, and J. Siegel, “Coordination of Communication: Effects of Shared Visual Context on Collaborative Work,” in Proc. CSCW ’00, pp.21-30, 2000.
5) C. Heath and P, Luff, “Disembodied Conduct: Communication through Video in a Multi-Media Office Environment,” in Proc. CHI ’91, pp.99-103, 1991.
6) D. Kirk, A. Crabtree, and T. Rodden, “Ways of the Hands,” in Proc. ECSCW ’05, pp.1-22, 2005.

* 対話者に共通する知識，信条，仮説．
† 対話によって，コモングラウンドを確立すること．

7) H. Kuzuoka, “Spatial workspace collaboration: a SharedView video support system for remote collaboration capability,” in Proc. CHI ’92, pp.533-540, 1992.
8) H. Kuzuoka, J. Kosaka, K. Yamazaki, Y. Suga, A. Yamazaki, P. Luff, and C. Heath, “Mediating Dual Ecologies,” in Proc. CSCW2004, pp.477-486, 2004.
9) H. Kuzuoka, T. Kosuge, and M. Tanaka, “GestureCam: A Video Communication System for Sympathetic Remote Collaboration,” in Proc. CSCW ’94, pp.35-43, 1994.
10) 岡本周, 酒田信親, 西田正吾, “光路を可視化したレーザポインタによる実物体指示,” 日本 VR 学会論文誌, vol.13, no.4, pp.421-428, 2008.
11) J. Ou, S. Fussell, X. Chen, L. Setlock, and J. Yang, “Gestural Communication over Video Stream: Supporting Multimodal Interaction for Remote Collaborative Physical Tasks,” in Proc. ICMI ’03, pp.242-249, 2003.
12) B. Scassellati, “Imitation and mechanisms of joint attention: A developmental structure for building social skills on a humanoid robot,” in Computation for metaphors, analogy and agents, vol.1562 of Springer Lecture Notes in Artificial Intelligence, Springer Verlag., 1999.
13) 山崎敬一 編, “実践エスノメソドロジー入門,” 有斐閣, 2004.

■S3 群 - 8 編 - 5 章

5-2 設計活動の支援

（執筆者：大平雅雄）[2009 年 2 月 受領]

ソフトウェア開発は一般的に，多人数の開発者がチームとして行う協調活動と見なすことができる．また，開発者のみならず，利用者（エンドユーザ），運用・管理者，経営者など様々なステークホルダ（利害関係者）がソフトウェア開発に参画する．立場や価値観の異なるステークホルダがそれぞれの関心事に基づいてシステムが満たすべき機能や特性を要求するため，要求同士が衝突したりステークホルダ間の意思決定や合意形成が困難になる場合が少なくない．そのため，ソフトウェアシステムを協調的に設計・開発するための方法論やグループウェアがこれまでに数多く提案されてきた[1]．

5-2-1 設計方法論

従来のソフトウェア開発では，ユーザは要求仕様の策定と検収時にのみ開発者と意見交換を行っており，要求仕様の欠陥が検収時に発覚し膨大な手戻り作業を発生させることが多かった．近年では，要求仕様や設計仕様の欠陥を防ぐとともに，システムを利用するユーザの立場や視点に立った使い勝手のよいシステムを設計・開発するために，ユーザを積極的に設計・開発プロセスに参加させる方法が採用されている．

（1）ユーザ参加型設計

ユーザ参加型設計（Participatory Design）は，開発するシステムを実際に利用することになるエンドユーザを，システムの設計段階からソフトウェア開発に参加させ設計過程やプロトタイプをこまめにチェックする機会を増やすことで，ユーザの要求（システムの機能や使いやすさなど）を満たしているかを常に確認しながらシステムを開発するための方法論である[2]．ユーザ参加型設計は，1960 年代から 1970 年代にかけてスカンジナビア諸国を中心に起こった労使関係の民主化運動が起源であるとされているため，スカンジナビアン・アプローチとも呼ばれる．

（2）JAD（Joint Application Design ／ Development）

ユーザ参加型設計が北欧を中心に盛んに行われるようになったのに対し，北米では 1970 年代後半に IBM 社で考案された JAD（Joint Application Design ／ Development）が普及した．ユーザを含むステークホルダを設計プロセスに積極的に関与させることにより仕様の策定を効率化することや仕様の品質を改善することを重視している点が，設計活動の民主性や利害対立の解消といった側面を重視しているユーザ参加型設計との大きな違いである[3]．JAD は後に，XP（Extreme Programming）や RAD（Rapid Application Development），適応型ソフトウェア開発などのアジャイル・ソフトウェア開発方法論の推進につながってゆく．

5-2-2 設計理由の記録モデル

設計理由（design rationale）とは，システムを設計した際になされた意思決定の理由や根拠を指す[4]．システムの仕様が策定されるまでには，ユーザがシステムを必要とする目的やその目的を達成するために必要な技術の比較や検討を通じて様々な意思決定が行われる．意思決定が行われた経緯を記録しておくことで，システム設計に関与した開発者だけではなく，プロジェクトに新しく参画する新人開発者や保守・運用担当者などが設計理由を参照し，シ

ステムを修正・変更する際に適切な処置を行うことが可能になる．また，設計理由の記録は，仕様変更の波及範囲や設計項目間の依存関係を特定したり，過去の議論の蒸し返しや未検討課題の放置を防ぐなど，システムの設計・開発において多くの利点がある．

(1) IBIS モデル

設計理由は後から容易に参照・理解できるように記録される必要がある．そのためのモデルの一つに IBIS（Issue-Based Information System）モデルがある．IBIS モデルは，問題点（Issue），問題に対する解答案（Position），解答案に対する議論（Argument），その他（Other）という 4 種類のノードを，支持（supports）や反対意見（objects-to）など 9 種類のエッジを用いて木構造として議論を構造化する手法である．IBIS モデルをシステム化した事例としては，非リアルタイムの議論支援システムである gIBIS[5] が有名である〔4 章 4-5 参照〕．

(2) QOC モデル

QOC（Question, Option, Criteria）モデル[6]は，問題（Question），解決案（Option），判断基準（Criteria）の 3 種類のノードからなる．ある問題に対していくつかの解決案があり，解決案はいくつかの判断基準によって評価される．ノード間は，各判断基準を解決案が満たすあるいは有利な場合は実線で，不利な場合は点線で結ばれる．IBIS と同様に木構造で構造化されるが，議論内容の構造化ではなく，最終的な設計結果と解決案の検討結果を簡潔に記録するためのモデルである．

5-2-3 ソフトウェア設計・開発支援ツール

ソフトウェアの設計・開発プロセスでは多人数の開発者の協調作業が必要とされるため様々なグループウェアが利用されている．特に，**ソフトウェア構成管理**（Software Configuration Management: SCM）には，開発プロジェクトの作業成果物の制御や管理，作業成果物に対する変更要求の管理・追跡など，多岐にわたる作業を支援するためのツールが利用される．

図 5･1 に示すように，ソフトウェア構成管理を支援するツールは主に，構成管理を支援するものと変更管理を支援するものからなり，作業成果物や作業履歴をはじめとする各種情報はリポジトリと呼ばれるデータベースに格納される．リポジトリ内の各種情報を関連づけることで開発チームが行う作業及び作業成果物を一元管理することができる．以下では，商用の支援ツールではなくオープンソースで開発され企業のソフトウェア開発においても利用されている無償の支援ツールを紹介する．

バージョン管理システムは，ソースコードや設計文書などの作業成果物の変更履歴（差分）を管理し，並行（ブランチ）開発や過去の任意のバージョンの復元，変更理由の参照，重複作業の防止を支援するシステムである．CVS（Concurrent Versions System）* や Subversion† が広く普及している．

バグ管理システムは，レビューやテストによって発見したバグを登録し修正状況を追跡・管理するためのシステムである．バグトラッキングシステム（Bug Tracking System: BTS）とも呼ばれる．Mozilla Foundation が開発しているウェブベースのバグ管理システム Bugzilla‡ や，

* CVS（Concurrent Versions System）: http://www.nongnu.org/cvs/
† Subversion: http://subversion.tigris.org/
‡ Bugzilla: http://www.bugzilla.org/

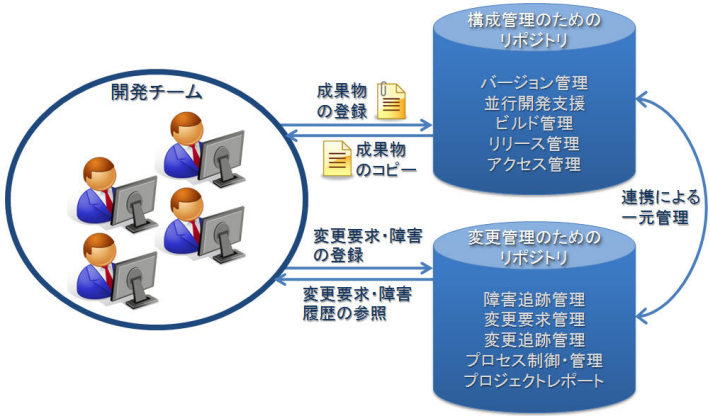

図 5·1　ソフトウェア開発における構成管理

日本製のオープンソースソフトウェアである影舞[§] などが有名である．また，バグ管理システムを発展させ，バグ管理だけではなくプロジェクト管理機能を併せ持つ Trac[¶]や RedMine[‖]なども最近注目されている．

ソフトウェア構成管理支援ツールのリポジトリに蓄積された情報を用いて，ソフトウェア開発時のアウェアネス情報（どの開発者がどのコードをいつ，どのように追加・修正したかなど）を共有し，円滑な協調作業を実現するための支援ツールの研究も盛んに行われている[7]．しかしながら，ソフトウェア構成管理支援ツールを含め，すべての開発者（支援ツールの利用者）がその意義や利用方法を十分に理解していないためにツールの効用を十分に享受できない事も少なくない．グループウェア全般に共通する課題でもあるが，ツール利用時の負担を軽減しすべての利用者がメリットを実感できるよう利便性を高めていくことは，オフショア開発など，分散開発の重要性が高まる近年のソフトウェア開発にとって重要な鍵になると考えられる．

■参考文献

1) 垂水浩幸, “グループウェアとその応用,” ソフトウェアテクノロジー シリーズ 12, 共立出版, 1999.
2) D. Schuler and A. Namioka, “Participatory design: Principles and practices,” Lawrence Erlbaum Associates, 1993.
3) J. Wood and D. Silver, “Joint Application Development,” John Wiley & Sons, 1995.
4) “Design Rationale: Concepts, Technique, and Use,” ed. by T.P. Moran and J.M. Carroll, Lawrence Erlbaum Associates, 1996.
5) J. Conklin and M. Begeman, “gIBIS: A Hypertext Tool for Exploratory Policy Discussion,” in Proc. ACM Conference on Computer Supported Cooperative Work (CSCW ’88), pp.140-152, 1988.

[§] 影舞: http://www.daifukuya.com/kagemai/
[¶] Trac: http://trac.edgewall.org/
[‖] RedMine: http://www.redmine.org/

6) A. MacLean, R.M. Young, V.M.E. Bellotti, and T.P. Moran, “Questions, options, and criteria: elements of design space analysis,” ed. by T.P. Morran and J.M. Carroll, Design rationale: concepts, techniques, and use, Lawrence Erlbaum Associates, pp.53-106, 1996.
7) M.D. Storey, D. Čubranić, and D.M. German, “On the use of visualization to support awareness of human activities in software development: a survey and a framework,” in Proc. 2005 ACM symposium on Software visualization (SoftViz ’05), pp.193-202, 2005.