

ネットワークを介した共同作業に対する支援環境の 参照モデルに関する一考察

落水 浩一郎

北陸先端科学技術大学院大学
情報科学研究科

〒 923-12 石川県能美郡辰口町旭台 1-1

Phone: +81-761-51-1260

Fax: +81-761-51-1149

e-mail: ochimizu@jaist.ac.jp

概要

本報告では、共同作業を通じて生成・変更されていく「中間成果物や決定事項とそれ等の状態」を、現実世界の状況を正確に反映しつつ管理する情報リポジトリの構成法を論じる。これを土台として、中間生成物や決定事項の矛盾や不確実さに満ちた状態を把握・減少させるための機能の実現が期待でき、さらに独立性や円滑性で特徴づけられる環境の快適性のある程度の達成が望める。

Consideration on a Framework for a Support Environment for Cooperative Works over a Computer Network

Koichiro Ochimizu

School of Information Science

Japan Advanced Institute of Science and Technology, Hokuriku
1-1 Asahidai, Tatsunokuchi, Nouni, Ishikawa 923-12, Japan

Phone: +81-761-51-1260

Fax: +81-761-51-1149

e-mail: ochimizu@jaist.ac.jp

Abstract

In this paper, we will discuss an structure of an information repository that keeps the states of artifacts/products and decisions, reflecting the state of progress precisely. With the help of the information repository, we can expect to construct functions to manage inconsistencies and uncertainties included in decisions and artifacts/products. Managing inconsistencies and uncertainties will also help us to achieve some degree of pleasantness of SDEs featured by independentness and smoothness

1 はじめに

資源共有、一貫性保持、安全性、耐故障性、負荷バランス等の問題を理想的に解決する分散仮想計算機構の存在を仮定しても、共同作業に従事する参加者間の地理的分散のために、依然として解決されるべき問題が残る。「共同作業の状態に関する、各自の状況把握のズレをいかに調整し、必要な情報を共有させるか」という問題はの一つである(図1)。例えば、決定事項とその状態の共有はコミュニケーションを円滑にするために重要であり、また中間生成物とその状態の管理と関係者間での共有は作業の進行を適切に制御するために必要である。

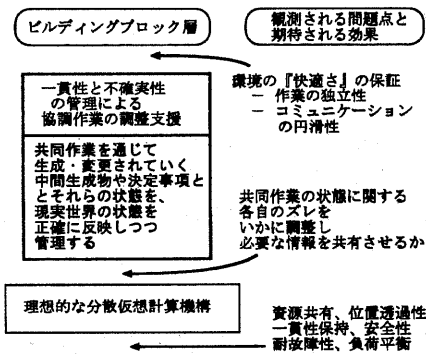


図1: 研究の動機

本報告では、共同作業を通じて生成・変更されていく中間成果物や決定事項とそれ等の状態を、現実世界の状況を正確に反映しつつ管理するための情報リポジトリに関して考察する。

このような情報リポジトリの存在は、中間成果物や決定事項に含まれる矛盾や不確実さを制御する機能を構築する土台を与える。さらに、矛盾や不確実さを制御することは、予測不可能性、非再現性、不完全性などの人間要因に基づく混乱を軽減させる一つの手段を提供しつつ、「快適な」ソフトウェア環境を特徴づける独立性や円滑性¹を達成する。

以下、第2節で情報リポジトリに対する要請を述

¹文献[7]において、筆者は、快適なソフトウェア開発環境が持つべき特性を明解性、独立性、円滑性、携帯性、増幅性の五つの観点から論じた。明解性とは「設計対象が評価可能であり、開発手順が明確かつ具体的であること」；独立性とは「必要なコミュニケーションを保証しつつも、個人の作業の独立性が保持できること」；円滑性とは「チーム構成員間のコミュニケーションが円滑であること」；携帯性とは「活動の場から、必要な情報に迅速にアクセスできること、さらには活動の場にそれを支える情報が追従できること」；増幅性とは、「ネットワーク上に遍在する、作業遂行に必要な情報にタイムリーにアクセスできること」を意味する

べ、第3節で、作業の責任範囲の管理、共有情報の変更管理、討議の筋道や決定の根拠の管理等の観点から構造化された情報リポジトリの構成について述べる。第4節では、図1に対するフレームワーク(「自在」CSCSDモデル)について述べる。

2 情報リポジトリに関する要請

ソフトウェア開発過程を「決定の積み重ね」であるとみなす。各自は自身の仕事の責任範囲において、みずから下した決定を管理する。例えば、ソフトウェア設計者は、設計対象に関する仕様を作成しプログラマに引き渡す。プログラマはその仕様に従ってデータ構造や制御構造を設計しコードを書く。決定(引き渡される仕様)は、引き渡された時点から、設計者とプログラマの間で共有される。引き渡されたものの内容の理解に関してギャップが存在する時に、様々な種類の会話が発生する。一般に、情報リポジトリの内容について各自は部分的な知識しか有しておらず、またその一部は他人と共有される。誰かが人々の間に、状況や知識についての理解のギャップが存在することに気付いた時、会話の契機が成立し、そのギャップを埋めるために会話が行なわれる。

そこで、ソフトウェア開発プロセスは、中間生成物の生成・変換プロセスにコミュニケーションプロセスを融合して定義されるべきである[5]。ソフトウェア開発における決定事項は2種類ある。一つは、設計者やプログラマによって作成される中間生成物や製品である。もう一つは、そのような中間生成物や製品を作成していく過程でなされる決定である。前者は、後者が複数個集まって構造化されたものである。

2.1 コミュニケーションに関する要請

後者の型の決定(デザイン・ラショナル)は伝達コストの安い会話によってなされる傾向がある。多くの研究者は、デザイン・ラショナルを管理するのに、会話内容そのものをその流れに沿って記録すれば良い、または、すべての会話内容を構造的に記録すれば良いという立場を取る。しかし、これは冗長である。なぜならば、このような会話は、内容をかみくみだいて説明したり、正確に内容を理解してもらおうための質疑応答を数多く含み、その内容は人の経験・知識・体調等に依存する。

例えば、結論にいたる速度、質は人によって異なる。早く達成した人が、ゆっくり、かみくみくよう

に、要点を「他人の能力」を勘案しつつ説明する状況が話し合いでは良く出現する。これを「かみくだく過程」と呼ぶことにする。かみくだく過程は、話者相互間で、理解の差や決定内容に関するくい違いが自覚され、それをなくそうとする努力がなされている場合にも生じる。

共同作業においては、参加者各自が、現在の状況が全体目標に対してどのような位置づけにあるのかを理解でき、進行状況や次の議論の目的(決定事項と未決事項)を共有できることが重要である。すなわち、会話の流れを円滑にするためには、

- ・ 決定事項の全体像の表現
 - ・ プロジェクトの目標に対する進捗状況
 - ・ ある話題に対する、部分的決定の積み重ねの構造とそれらの決定の際に存在したさまざまな前提条件
 - ・ 一つの部分的決定に関する確信度と手間
- のような情報が必要になる。

これらの要請は、決定事項とその状態を、その全体像から詳細レベルにいたる決定の粒度で、系統的に管理できるリポジトリの必要性を示唆している。

ところで、リポジトリに記録される決定は通常、矛盾があり不確実である。すなわち、「完全なものを一度には作成できない」という人間の特性に起因して、決定事項はしばしば変更される。多くの場合、決定は一時的なものであり、時間がたち、状況が変化するにつれて、合意が不合意になり、不合意が合意に変化することはしばしば経験されることである。矛盾や不確実さの存在を前提とした作業を支援するような環境の存在が望まれる。

2.2 ソフトウェアプロセスに関する要請

人間は完全なものを一度には完成できないので、繰り返しが発生する。その度合は、ソフトウェアの規模が増大するにつれて大きくなる。また、人間は機械ではないので、その振舞いを予測し再現するのが困難である。ソフトウェア・プロセス・モデリングにおいては、これらの人間要因を考慮に入れる必要がある。

また、個人の活動とチームの目標の間の「調整」の問題もある。例えば、他人のやった仕事を引き継ぐ。他人に仕事を引き渡す。社内やプロジェクト固有の規則を順守する。一定の品質が保証されたプロダクトを生成するなどの例がある。最も重要なことは、これらすべてのことを納期にあわせて、限られた時間資源で実行しなければならないことである。個人の目

標とプロジェクトの目標の間にはひらきがある。前者は、「自身の能力の範囲内で納得しながら良い仕事を楽にする」という自己中心の仕事のタイプを容認し、後者は、「限られた時間範囲内にすべての仕事を終了する」という自己犠牲、献身型の仕事のタイプを要求する。双方を同時に達成できる人もいるがそれは稀な例である。個人の目標とプロジェクト・チームの目標をつなぐことができる調整支援のためのプロセス・モデルが必要である。

我々のアプローチでは、手順を書き下すことではなく、仕事の静的/動的コンテキストを定義することに力点を置く。試行錯誤は自然なものであるとして、達成目標(品質基準)、犯してはならない制約や規則、他人との仕事のつながりなどを各自の作業環境のまわりに配置する。

3 共同作業の状態を反映した、決定管理のための情報リポジトリ

ソフトウェア開発においては、ある中間生成物が入力され、既存の成果物、標準規約、品質基準等を参照しつつ、ツールによる変換や人間の思考によって、出力となる中間生成物や製品が生成される。すなわち、中間生成物や製品の間には入出力に関する依存関係が存在する(図2)。粗粒度レベルの依存関

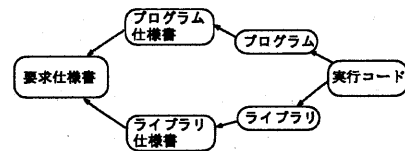


図2: 粗粒度レベルの依存関係の一例

係は、仕様書や設計書などの文書間の入出力依存関係を定義する。我々は、中間生成物間の粗粒度レベルの依存関係をソフトウェア・プロセス・モデルの定義とコミュニケーションの記録を統合する基礎として採用する。もちろん、中間生成物間の粗粒度レベルの依存関係が上記の機能実現の土台として必要十分であるとはいいいがたい。しかし、変換作業の実行順序の設計にあたって制約を与え、会話の発生順序に関する概略を与えている意味において、統合の主要な土台の一つであると考えられる。

3.1 仕事の責任範囲をタスクとして定義することについて

共同作業者の責任範囲の定義という立場からタスクの定義を導入する。タスクは中間生成物や製品(以後、中間成果物オブジェクトと呼ぶ)の部分集合を、チーム構成員各自の役割に応じて割り当てたものである(図3)。それぞれのタスクは、例えば、割り

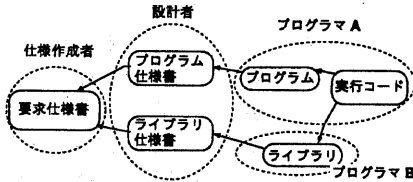


図3: タスク割り当ての例

当てられた中間生成物や製品の記述に関する様式や規約、作業遂行上の制約、他人との仕事の関係、品質基準などの属性を持つ。属性定義の詳細は個々の組織やプロジェクトに依存する。タスクの束構造としてソフトウェア・プロセスを定義する(図4)。我々

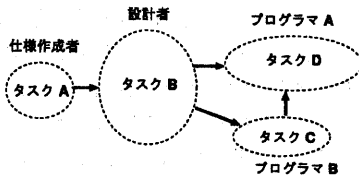


図4: ソフトウェアプロセスの一例

は、図5に示すような共有情報の管理機構をすでに開発した[4]。そのプロトタイプ「群舞」は、分散作業空間管理者オブジェクトと自律メディエータ・オブジェクトからなり、特徴は以下の通りである。

1. 分散作業空間管理者オブジェクトは、ソフトウェア開発者各自の仕事の責任範囲(分散作業空間、図4におけるタスクに対応)を規定し、共有された中間成果物オブジェクト(分散作業空間の積集合)の変更管理を支援する
2. 共有作業空間に属する中間成果物オブジェクトはそれぞれアクセスリストを持ち、許可されていないアクセスに対しては作業空間管理者オブジェクトを介して自律メディエータ・オブジェクトを起動し、関連者間での一時的な権限の委

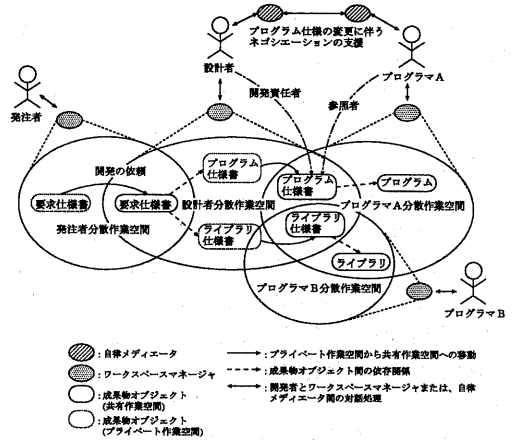


図5: 責任範囲と共有情報の管理

譲に関するネゴシエーションを支援しつつ「協調書き込み[4]」を実施する。

それぞれのオブジェクトは、対応するメタ・オブジェクトを持ち、状況に応じて適切な振舞いを動的に選択することを可能にする。中間成果物オブジェクトは図6に示す状態遷移図をそれぞれ内蔵することで、中間成果物の進捗状況を反映させる。ソフトウェアプロセスの実行には終りがないので、状態遷移図には最終状態がない。

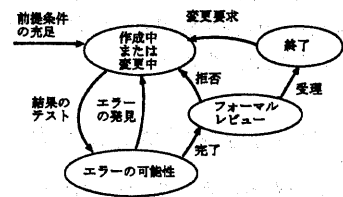


図6: 中間生成物に付加される状態遷移図

3.2 コミュニケーション内容の記録法

3.2.1 討議の筋道と根拠を記録することについて

会話の流れは会話参加者の関心の推移を表現する。そこでは参加者の間で最も関心の高い事項から会話が始まる。その会話の前提条件の欠落に途中で気付く、一時話題がそちらに移ることもある。ある話題について話が一段落した時、続いて話合うべきことが話題として持ち上がる。一度の話し合いでは結論

に達せず話を中断することもよくある。この時、別の話題に話が移った後で、突然、あることに気付いて、話題がもとに戻ることもある。このような会話の流れの中で決定されていく事項をどのように記録すれば良いのだろうか。

我々は、話し合いの途中で、そこまでの話を整理し、解決された問題とそうでないものを整理した上で次に話し合うべきことを決定するような場面をよく経験する。この時、我々は以下のような情報を整理していることが多い。

- ・すでに決定されたことと、これから決定されるべきことに対する全体像
- ・全体像に対する、現状の位置づけ
- ・最終目標に対する進捗状況
- ・次に話し合うべき話題

このような事柄は当然記録の対象とすべきであろう。ところで、議論が進むにつれてそのような整理結果が変化することも考慮する必要がある。さらに、いったん決定したことが不完全であることに後になって気付くこともしばしばある。討議結果の再構成や蒸し返しを支援できる情報も記録の対象に入れるべきである。そのためには以下のような情報を記録しておく必要がある。

- ・ある話題に対する、部分決定の積み重ねの構造とそれらの決定にあたって必要なさまざまな前提条件
- ・一つの部分的決定に関する確信度と手間

さらに、話し合いの話題はソフトウェア・オブジェクトに起因するので、中間生成物や中間生成物間の依存関係との関連も当然記録の対象とすべきである。

会話を通じてなされる決定事項を、図 7 に示すように、討議空間、討議プロセス、討議の型からなる 3 層記録スキーマで管理する [5]、[8]。

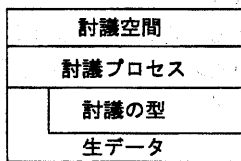


図 7: 会話を通じてなされる決定事項の管理法

3.2.2 討議空間

それぞれの話題に対する決定事項間の因果関係を討議空間で表現する。その役割は、決定事項の全体像と最終目標に対する現在の議論の適切な位置づけ

を我々に提供することにある。討議空間は、後に定

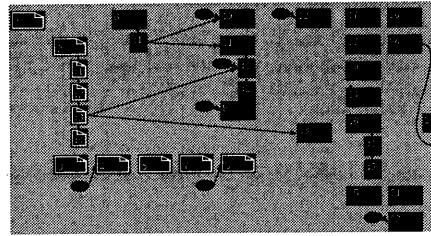


図 8: 討議空間

義する討議プロセスのグラフであり、節は討議プロセス、枝は討議プロセス発生の原因を表現する。図 8 に討議空間の一例を示す。長方形の箱は討議プロセスを表わし、正方形の箱は、長方形で表現された討議プロセスを遂行するために発生した子討議プロセスを表わしている。枝には三つの型がある。

継続 ある討議プロセスが、別の討議プロセスの完了に伴ない発生する

詳細化 ある話題に関する討議を完結するためには、いくつかの部分話題を討議する必要がある。

相互関連 二つの討議プロセスが、ある内容を共有して並行に走行している。

我々はプロトタイプ「栗」を開発したが、その詳細については文献 [5]、[6] を参照されたい。

3.2.3 討議プロセス

同じ話題に対する会話をグループ化するものとして討議プロセスを導入する。いくつかの話題に対するそれぞれの討議内容は通常まざりあっている。ある話題に対する一連の会話を討議プロセスで管理することにより、討議の参加者がそれぞれの話題に対応する討議の進捗と筋道を把握できることは有用である。また、ある話題に対する、部分決定の積み重ねの構造と、それらの決定に関するさまざまな前提条件を管理することも支援する。

討議プロセスは、図 9 に示す状態遷移図にしたがって推移する。討議プロセスは、それぞれの話題に対して生成され、討議の期間中生存する。討議はいくつかの前提条件が満たされたとき開始される。前提条件の例として必要な資料の準備、関連する人々の参加、指定された時間の到来などが挙げられる。いくつかの会話が状態「討議」においてなされる。権

利を持つ誰かが宣言することで、討議は終了し、結果(討議を通じてなされた決定)が文書化される。

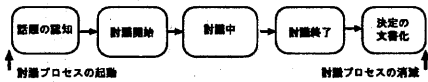


図 9: 討議プロセスに付随する状態遷移図

3.2.4 討議の型

田中等 [9] による、会話の型に関する研究成果に基づいて、我々は会話の型を：伝達と整合、決定、創造の三つに分類した。会話中の一連の発言を伝達・整合、決定、創造などの討議の型で管理する²。例えば、良く知っている人がそうでない人に決定事項や知識を「伝達」する。お互いの言い分を聞きながら、いくつかの代替案の中から一つの案を「選択・決定」する。部分的な知識を持つ人々が、それらを統合して新しい情報を「創造」する。討議の型の役割は、一つの部分的決定に関する確信度と手間を記録することにある。伝達・整合、決定、創造などの討議の型におけるどの状態で会話が中断・終了したかにより、確信度や完遂度を判定できる。時間属性や発言の回数は決定に要したコストを表現できる。

伝達・整合の図式表現

「伝達・整合」の目的は、質疑応答を繰り返すことにより、情報や知識が共有された状態に達することである。図 10 の図式表現を状態遷移図により示す。まず、アイデア、指示、生産物の提示・引き渡し等に

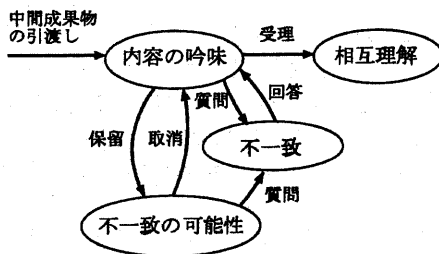


図 10: 伝達・整合

駆動されて会話ははじまる。受け手はその内容を吟

²それぞれは会話の筋道を記録する上で基本的かつ重要なものであるが、もちろんすべての会話の型をカバーするわけではない

味する。もし、受け手が内容を理解した場合は、それを受理し、送り手と受け手は「相互理解」の状態になる。もし受け手が態度を鮮明にしない場合には「不一致の可能性」がある。そこで、受け手が何らかの質問を送り手に対して発する。この段階で送り手と受け手は「不一致」の状態になる。ここで、送り手と受け手の間でギャップを埋めるための努力が、例えば、いくつかの質疑応答を通じて開始される。会話は「受理」または「不一致」のどちらかの状態で終了するだろう。前者は伝達・整合の成功を意味し、後者は伝達・整合の失敗を意味する。

例 1: 生産物の引き渡し

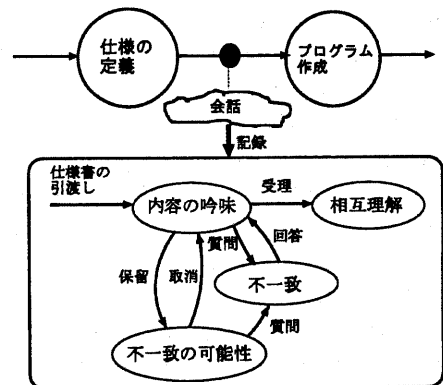


図 11: 生産物の引き渡し

ソフトウェア設計者は仕様を定義しそれをプログラムに渡す。この時、仕様内容について共通理解を達成するためにコミュニケーションの初期の段階で伝達・整合の会話が発生する(図 11)。何が合意され何が合意されていないかを明確に記録しておくことは有用である。通常、この型の情報は生産物の引き渡し後には消滅することが多く、種々の問題を引き起す場合が多い。

決定の図式表現

「決定」の目的は、図 12 に示すように、解に対するいくつかの候補の中から、適切なもの一つを選択することにある。この型の会話は解に対する複数の解候補が存在し、選択・決定が必要であることから発生する。その意味で、初期状態は「複数の解候補の存在」とする。参加者の内誰かが、議論の対象範囲を狭めるため、理由と共に意見を開陳する。その

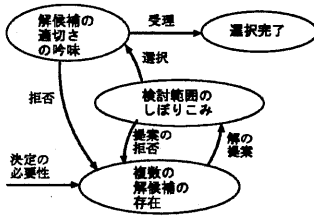


図 12: 決定

意見を吟味した後、ある場合には、それが拒否され、またある場合にはそれが受理される。前者の場合はさらに他の意見が表明されるだろう。いくつかの議論を経た後、会話は収束し、候補の内のどれか一つが解として選択されることになる。

例 2：ソフトウェアレビュー

ソフトウェアレビューのプロセスはソフトウェア開発者とレビューアの間で意思決定のためのコミュニケーションプロセスである(図 13)。まず初期には、

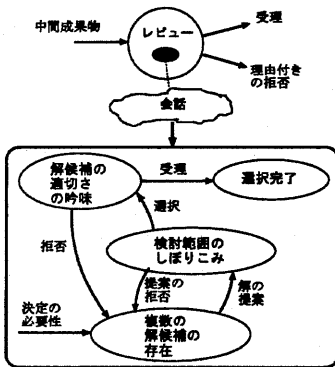


図 13: ソフトウェアレビュー

「受理」か「拒否」かの二つの可能性がある。意見はレビュー対象物のすべての欠陥を明示的に洗い出す形で表明される。もし結果が拒否であった場合は、これらの意見の内のいくつかがレビュー対象が拒否された理由と記録される。

創造の図式表現

「創造」の目的は何か新しいものを創ることにある。「創造」に対する図式表現は「伝達・整合」と「決定」を合成することで図 14のように表現される。

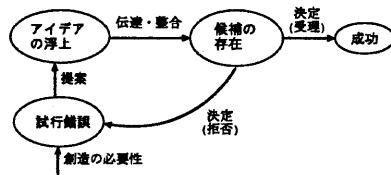


図 14: 創造

討議の型の組み合わせ

「決定」と「伝達・整合」の組み合わせ

決定のあるものについては、チーム構成員の主要メンバーによってなされることがある。この場合には、「決定」後、数多くの「伝達・整合」の会話を利用して、全体的コンセンサスを得るための努力がなされる。

「伝達・整合」と「決定」の組み合わせ

最初は伝達・整合の目的で開始された会話が、送り手と受け手の間での厳しい対立が発生し、会話は「伝達・整合」のみによっては終了せず、「決定」の型の会話が開始される。この時、「伝達・整合」の会話の状態は「不一致」の状態に留まる。

3.2.5 グループウェアベース

上記スキーマを持つリポジトリを、我々はグループウェアベースと呼ぶ。グループウェアベースは、図 15に示すような管理ページの集合である。管理ページは各討議プロセスごとに作成される。図 15の詳細については文献 [5] を参照されたい。

1	プロセス ID	12	討議履歴(討議の型の系列)
2	話題	13	終了時刻
3	討議の目的	14	達成度
4	討議開始条件	15	決定事項の格納時刻
5	議論の蒸し返し	16	決定
6	進行役	17	生データ
7	終了判定者	18	親プロセス ID
8	参加者	19	子プロセス ID
9	予定討議時間	20	継続プロセス ID
10	起動時刻	21	並列プロセス ID
11	開始時刻		

図 15: 管理ページ: グループウェアベースの記録単位

4 共同作業に対する参照モデル

前章までに検討した環境構成の諸要素を図 16に示すように、階層構造として構造化する。

ユーザインタフェース (最上層): 分散サーバ、グループウェア・ベース、プロセス・サーバの状態を表示すると共に、Ellis が指摘した各種情報 [3] を必要に応じて表示する予定である。

ツールキット層 (第 2 層): CASE tool やグループウェアおよびそれらの統合機能が配置される。

矛盾、不確かさの制御層 (第 3 層): deviation[2] による矛盾管理機能、「決定事項の不確かさ」の計算機能などの機能を配置する予定である。

各種サーバ (第 4 層): 分散サーバ、グループウェア・ベース、プロセス・サーバを置く。プロセス・サーバは、タスクとその実行順序、消費可能資源等を定義する。

タスク管理層 (第 5 層): タスクとその属性の管理

中間生成物間の依存関係の管理 (第 6 層): 中間生成物間の依存関係の管理

CSCSD サーバ (最下層): 既存の分散システムとの接続を支援する

ユーザ・インタフェース (状況の表示)		
CASEツールとグループウェア		
決定事項の一貫性と不確かさの管理		
プロセス実行管理 (プロセスサーバ)	共有情報の変更 管理 (分散サーバ)	決定の筋道と根拠の 記録 (グループウェア ベース)
責任範囲 (タスク) の定義		
粗粒度レベルの依存関係の管理		
分散計算システムとの結合		

図 16: CSCSD モデル

5 今後の課題

「はじめに」で述べた目標に近づくための次の研究課題は以下の通りである: 分散に依存する問題のより精密な把握; 決定事項の不確かさのファジイ理論による定量化; 討議空間の自動生成; 分散作業空間のモバイル環境上での実現; 分散開発方法論の定義謝辞

ミラノ工科大学 Carlo Ghezzi 教授による「一貫性管理」に関する示唆に心より謝意を表する。

参考文献

- [1] R.M.Adler and R.C.Paslay, "Design of Distributed Systems", Encyclopedia of Software Engineering, John Wiley and Sons.
- [2] G.Cugola, E.Di Nitto Frank, C.Ghezzi: "How to deal With Deviations During Process Model Enactment", Proc. of ICSE 17, pp.265-273, April 1995.
- [3] Clarence Ellis, Jacques Wainer: "A Conceptual Model of Groupware", Proc. of CSCW 94, pp.79-88, 1994.
- [4] 堀, 落水: 「ソフトウェア開発における自己反映オブジェクト指向モデルに基づく共有情報の管理法」, コンピュータ・ソフトウェア, Vol.13, NO.1, pp.37-54, 1996. M.Hori, Y.Shinoda, K.Ochimizu: "Shared Data Management Mechanism for Distributed Software Development Based on a Reflective Object-Oriented Model", LNCS 1080, Advanced Information Systems Engineering, pp.362-382, 1996.
- [5] 門脇, 落水: 「ソフトウェア・プロセス実行における系統的コミュニケーション支援の一方式」, Jaist Research Report, IS-RR-97-0009S, ISSN 0918-7553, March 1997.
- [6] 近野, 門脇, 落水: 「グループウェアベース「乗」を用いた電子会議内容の進捗把握と文書化の支援」, 情報処理学会ソフトウェア工学研究会資料, 107-12, 1996.
- [7] 落水: 「ネットワークを介した共同作業の支援環境構築にむけて」, SEAMAIL, Vol.10, NO.9-10, pp.2-37, 1997.
- [8] K.Ochimizu and C.Kadowaki: "A Framework of a Support Environment for Cooperative works over Distributed Computing System based on a Decision Management", Jaist Research Report, IS-97-0013S, ISSN 0918-7553, March 1997.
- [9] 田中, 荒木, 増田: "対人的コミュニケーションにおける電子的メディアの特性と効果", 情報処理学会グループウェア研究会資料 4-8, pp.53-60.