

UML ダイアグラムとチャットを関連付けた 分散共同モデリング支援システム

徐 冬 梅^{†1} 黒 木 純^{†2}
大 瓶 佳 秀^{†1} 樋 山 淳 雄^{†2}

ネットワーク技術の進展とビジネスの国際化により、ソフトウェアの分散開発は一般的になってきた。このような背景から、分散共同モデリング支援システムが提案されてきた。本研究では従来の分散共同モデリング支援システムが扱っていない2つの側面に着目する：1) モデリング時におけるダイアグラムとそれに対する会話内容を関連付けることの有効性、2) 見直し時におけるダイアグラムとそれに対する会話内容を関連付けることの有効性。著者らは上記2つを実現した UML ダイアグラムを対象とした分散共同モデリングシステム *Libra-on-Chat* を開発した。実験の結果、被験者は議論中のモデル要素に注目することができることが明らかになった。また、見直しの際、設計根拠の把握に有益であることも示された。

A Distributed Collaborative Modeling Support System by Associating UML Diagrams with a Chat Tool

DONGMEI XU,^{†1} JUN KUROGI,^{†2} YOSHIHIDE OHGAME^{†1}
and ATSUO HAZEYAMA^{†2}

With the advancement of network technology and globalization of business, distributed software development has become in common. Along with this trend, distributed synchronous collaborative modeling supporting systems were proposed. This study points out two issues traditional systems have; 1) they don't relate conversations with elements of a model diagram, and 2) they can't replay histories of modeling by associating contents of conversations with elements of a model diagram. The authors developed a distributed synchronous collaborative modeling supporting system for UML diagrams, called *Libra-on-Chat*, which solved the issues. As the result of experiments, the system allows participants to focus on elements in a diagram under discussion. It also enables developers to grasp design rationale in a diagram.

1. はじめに

ネットワーク技術の進展にともない、企業の国際化が急速に進んでいる。ソフトウェアの分散開発も一般的になってきた。しかし、ソフトウェアを分散環境下で共同開発する場合、開発者間のインフォーマルコミュニケーションの不足が共同開発を難しくしている¹⁾。特に、要求分析や設計等の上流工程の作業はコミュニケーション集約的な作業である。ソフトウェア開発におけるコミュニケーションには意思決定や設計根拠に関わるものが含まれ、また成果物と密接に関わる⁹⁾。上流工程の作業成果はダイアグラムを用いて表現されることが多い。近年主流になってきたオブジェクト指向によるソフトウェア開発では、上流工程での成果物を UML (Unified Modeling Language)⁸⁾ で定義された数種類のダイアグラムを用いて作成する。一方、対面会議は出張等にもなうコストが発生する。分散環境下におけるコミュニケーションをともなうモデリング作業を支援するために、同期分散開発環境における共同モデリング支援システムが提案されてきた^{1),3)}。しかし、これらのシステムは成果物作成支援に対して以下の問題点がある：

(1) モデリング時において、コミュニケーションツールと成果物作成ツールが連携していない

RD-UML¹⁾ は UML を対象とした同期モデリング支援システムであるが、利用しているテキストコメントツールは成果物作成ツールと密接に関連付けられておらず、コメントは UML モデル要素と関連付けられていない。

Elapiz は UML ダイアグラムの共同編集を行うことができるシステムである。また、コミュニケーションも支援している³⁾。しかし、コミュニケーションは生成されたモデル要素に対してのみ行うことができ、ダイアグラム作成前の会話内容を保持することができない。また、モデリング作業におけるコミュニケーションにはモデル要素に直接関連付けられないものも存在する。さらに、コミュニケーションは Bulletin Board System (以下、BBS) による非同期コミュニケーションのみをサポートしている。分散環境下でモデリングをリアルタイムに行う場合にコミュニケーションが非同期であることは現実的でない。

コンピュータヒューマンインタラクション (CHI) の分野において、コミュニケーション

^{†1} 東京学芸大学大学院教育学研究科
Graduate School of Education, Tokyo Gakugei University

^{†2} 東京学芸大学教育学部
Faculty of Education, Tokyo Gakugei University

ツールを成果物の特定の部分に関連付けて、会話をリアルタイムに行えるシステムが提案されている^{2),5)-7)}。

Anchored Conversation はワードプロセッサのドキュメント中に注釈を付与することを可能とするツールである²⁾。しかし、共有する成果物を同期で共同編集することはできず、分散環境下において同期でコミュニケーションを行いながら UML ダイアグラムを共同作成することを支援するには適していない。

Lock-on-Chat は共有した画像の任意の場所にチャットを関連付けてコミュニケーションを可能にするシステムである⁶⁾。しかし、共有されている成果物である画像を同期で共同編集することはできない。

成果物として写真をリアルタイムで共有し、チャットによるコミュニケーションを可能にしたシステムがいくつか開発されている^{5),7)}。

xExplorer は地図情報と位置情報を活用した携帯端末ガイドシステムで、複数人が同一エリアを別々に行動し、写真のリアルタイム共有とチャットによるコミュニケーションを可能にするシステムである⁵⁾。しかし、写真とチャットを連携させることはできない。

PhotoChat は写真のリアルタイム共有とペンインタフェースにより手書きの注釈をリアルタイムに付与できるシステムである⁷⁾。さらに、複数の写真の間にハイパーリンクを張れるようになっている。

(2) 見直し時において、モデリング時に行ったモデル要素と会話内容の関連付けを時系列に再現することができない

設計作業において、振り返って見直したときに設計根拠⁴⁾が必要になることがある。また、同期モデリング作業に欠席者がいる場合、設計根拠を伝達することが必要になることもある。しかし、従来の同期分散開発環境における共同モデリング支援システムはこれらの会話内容を UML ダイアグラムと関連付けて管理していないため、このような要求に応えることは困難である。本研究が対象とする設計活動におけるコミュニケーションでは、意思決定につながる議論がどのようになされてきたかを把握するために、会話の時系列の推移が重要となる。CHI の分野で研究開発されたコミュニケーションを画像や写真等の成果物に関連付けたシステムは、会話内容の時系列な推移について主眼は置かれていないようである。PhotoChat では写真の撮影時刻はシステムの閲覧画面にも表示されているが、コミュニケーションについて時刻情報は明示的に表示していない。実験結果からも利用者は会話内容の時間の推移には着目していないようである。

本研究では、この 2 つの側面から、UML ダイアグラムを共有、共同編集可能で、かつ

UML ダイアグラムの特定モデル要素に会話内容に関連付けて管理する、分散共同モデリング支援システム Libra-on-Chat を開発する。本システムは、会話内容を時系列に保存し、設計根拠として参照することもできる。

本論文の構成を以下に示す。2 章では本研究が想定する分散共同モデリングプロセスを提案する。3 章では分散共同モデリングプロセスを支援するために開発したシステムについて述べる。開発したシステムの有効性を検証するために行った実験ならびにその結果について 4 章で述べ、最後に結論と今後の課題を述べる。

2. 分散共同モデリングプロセス

本章では、本研究が想定する分散共同モデリングプロセスについて述べる。図 1 はそのプロセスを図示したものである。モデリングプロセスは、開発者個人のモデリング作業と分散同期共同モデリング作業から構成される。

(1) 個人のモデリング作業

このステップでは、各開発者が先行文書（たとえば、要求仕様書）を参照しながら、個々に UML ダイアグラムを作成するステップである。これは会議において配布資料を事前に作成し、それに基づいて議論を進めていくことがあるのと同じように、分散同期共同モデリン

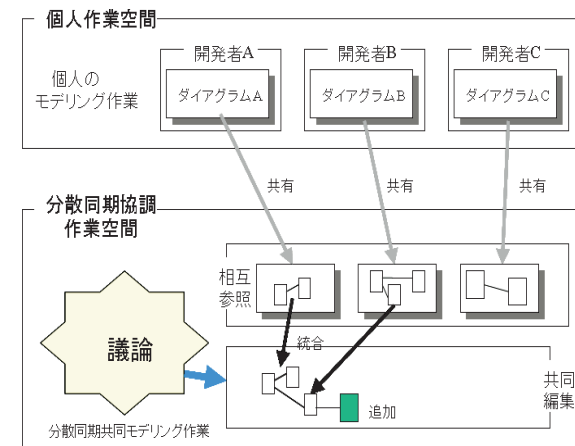


図 1 分散共同モデリングプロセス

Fig. 1 Distributed collaborative modeling process.

グ作業に先立ち開発者各自が事前に検討した内容をダイアグラムとして作成する場合を念頭に入れている。ただし、このステップを省略し、分散同期共同モデリング作業を行うことも想定している。

(2) 分散同期共同モデリング作業

このステップでは、開発者が議論を行いながら共同作業の成果であるダイアグラムを作りあげる。その過程において、各開発者が(1)のステップで作成したUMLダイアグラムを同期環境で共有・相互参照し、必要に応じてダイアグラムの要素(クラス、属性、関連等)を共同編集中のダイアグラムに統合したり、モデル要素を新規に追加したりして、ダイアグラムを作成する。

3. Libra-on-Chat : 分散共同モデリング支援システム

2章で述べたモデリングプロセスを支援するシステムを開発した。本章ではシステムアーキテクチャと主要機能について述べる。図2にシステムアーキテクチャを示す。

システムはクライアントサーバ方式で実装されている。UMLエディタはPCでユースケース図、クラス図、シーケンス図を作成することができ、作成したダイアグラムのファイ

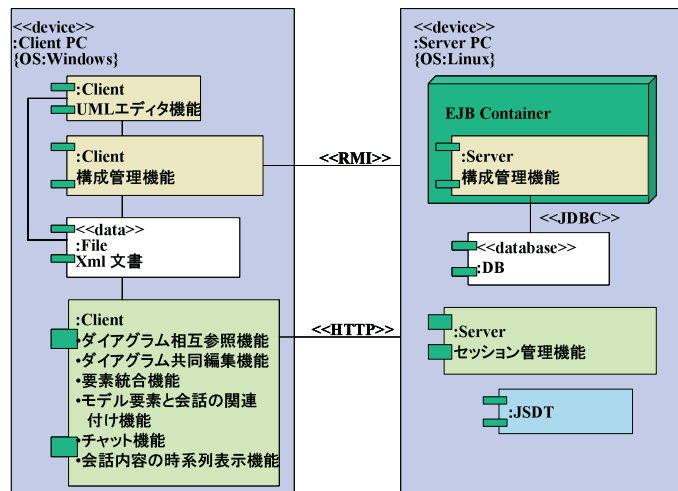


図2 システムアーキテクチャ
Fig.2 System architecture.

ルを当該PCに保存することができる。後述するダイアグラムの相互参照、ダイアグラム要素の統合、議論等のクライアント間での同期通信はSun Micro Systemsが開発したJava Shared Data Toolkit (JSDT)¹⁰⁾を用いて実現した。Libra-on-Chatは開発者が個人で作成したUMLダイアグラムを共有し、会話内容を同期で作成したUMLダイアグラムの特定のモデル要素に関連付け、それらを参照しながら、UMLダイアグラムを作成するシステムである。以下でシステムの主要機能について説明する。

(1) ダイアグラムの相互参照

開発者は自身が作成したUMLダイアグラムを他の開発者に公開することができる。「共有ビュー」(図3a)を開くことにより、それを可能にする。2人以上のダイアグラムを相互参照するには、タブによりダイアグラムの表示を切り替えることにより行う。

(2) ダイアグラムの共同編集

「共同モデルエディタ」(図3b)において、開発者が同期で協調してモデル要素を追加、削除、編集することができる。ある開発者によるこのエディタ内での操作が、瞬時に他の開発者の共同モデルエディタに反映される。開発者がエディタ上で新規にモデル要素を追加す

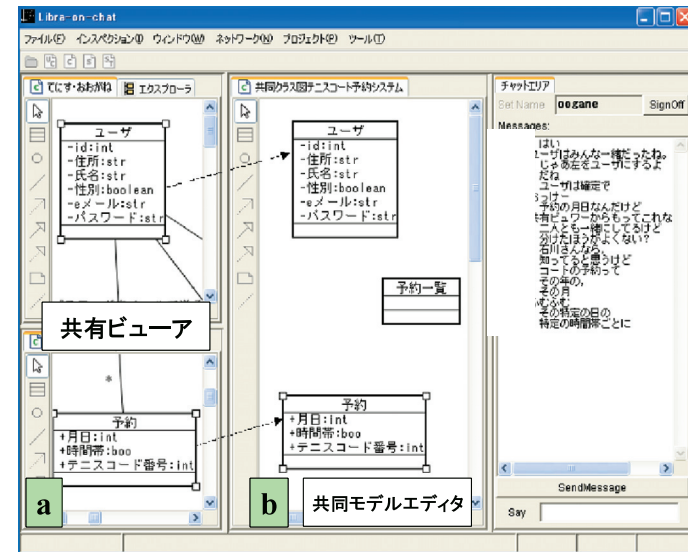


図3 ダイアグラムの相互参照と共同編集機能の画面イメージ

Fig.3 Screen shot of the cross-reference function and collaborative editor in Libra-on-Chat.

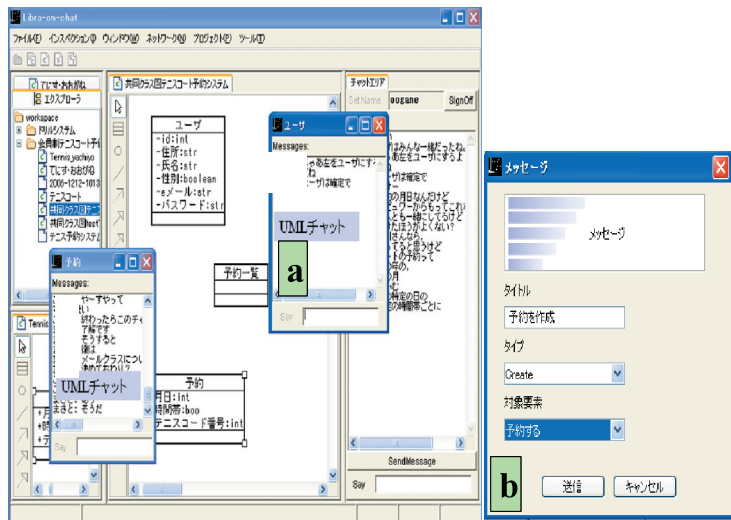


図 4 UML チャットとモデル要素への関連付け
Fig. 4 UML chat and its association with a model element.

ることできるし、「共有ビューア」に表示されているある開発者が作成したモデル要素をこのエディタに統合することも可能である。これにより個人の作業成果を同期共同作業に利用することが可能になる。そして、「共同モデルエディタ」上のモデル要素に対する編集や、モデル要素の削除を行うことも可能である。

(3) UML モデル要素と会話内容の関連付け

本機能は、特定のモデル要素に対する議論を支援するものである。

共同モデルエディタでモデル要素もしくは共同モデルエディタの背景を選択後、メニューを選択することによりチャットウィンドウが開かれる(図 4a。本システムでは「UML チャット」と呼ぶ)。UML チャットで会話を終了させるアクションを起こすと、会話内容をどのモデル要素に関連付けるのか等を指定するダイアログが表示される(図 4b)。ここで、UML チャットの属性として、タイトル(後述の会話内容の時系列表示機能で使用する)、議論の種類(モデル要素作成の議論か、修正の議論か、モデル要素削除の議論かを指定する)、関連付けるモデル要素を指定する(モデル要素を指定しなければ、特定のモデル要素に関連付かない会話として記録される)。すると、図 5 に示すアイコンとして、会話内容がモデル要

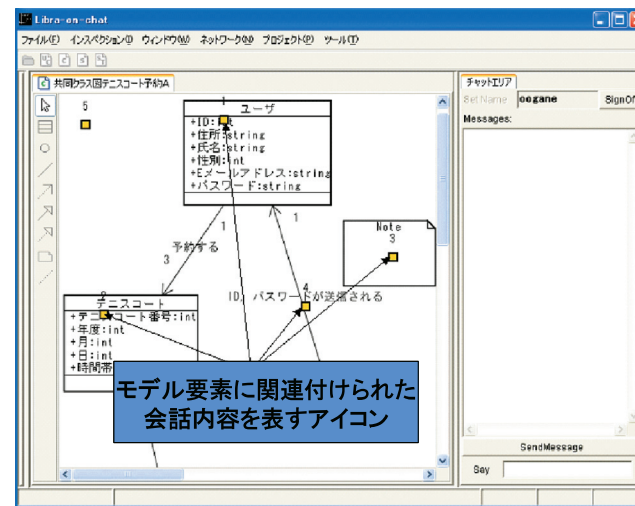


図 5 モデル要素に関連付けられた会話内容
Fig. 5 Conversations associated with model elements.

素に関連付けられる。関連付けのタイミングや誰が関連付けるのかは議論の中で決めることを想定しており、システムとして制約を課していない。アイコンをダブルクリックすることにより会話内容とその時刻情報を閲覧することができる。1つのUMLモデル要素に複数の「UMLチャット」を保存することができ、アイコンに付けられた番号により時系列関係を表す。

UMLチャットは関連付けられているモデル要素そのものとの関連を保持しているので、モデル要素の名前の変更や、属性の追加や削除があっても影響を受けることなくモデル要素と会話内容の関連付けは維持される。また、モデル要素が削除された場合には、そのモデル要素に関連付けられていたUMLチャットは削除されずにモデル要素と関連付けられないUMLチャットとなる。ただし、現在の実装ではモデル要素の変化ログを保持していないので、モデル要素そのものが削除されてしまった場合にUndoを行うことができない。UMLチャットはモデル要素の設計根拠として利用することができる。

また、UMLチャットのアイコンを右クリックすると図 4bのダイアログウィンドウが現れ、UMLチャットの属性(タイトル、議論の種類、関連付けるモデル要素)を修正することができる。

(4) 同期モデリング作業の全体像を把握するためのチャット

特定のモデル要素に関連付かない議論を行う場である（図 3 右端欄．本システムでは「チャット」と呼ぶ）．このエリアで行った議論を UML モデル要素に関連付けることも可能である．会話内容をモデル要素に関連付けたいタイミングで SendMessage ボタンを押下すると図 4b と同じダイアログウィンドウが現れるので，UML チャット同様，チャットの属性を指定することによりモデル要素に関連付けることができる．

(5) 会話内容の時系列表示

チャットの内容を時系列に表示するものである（図 6）．UML チャットに付けられたタイトルと番号により内容を識別する．タイトルをクリックすることで，そのメッセージはどのモデル要素に関連付けられていて，誰が，いつ，何を議論したのかを閲覧することができる（「メッセージ内容」というタブのところに表示される）．これにより，共同モデリングの過

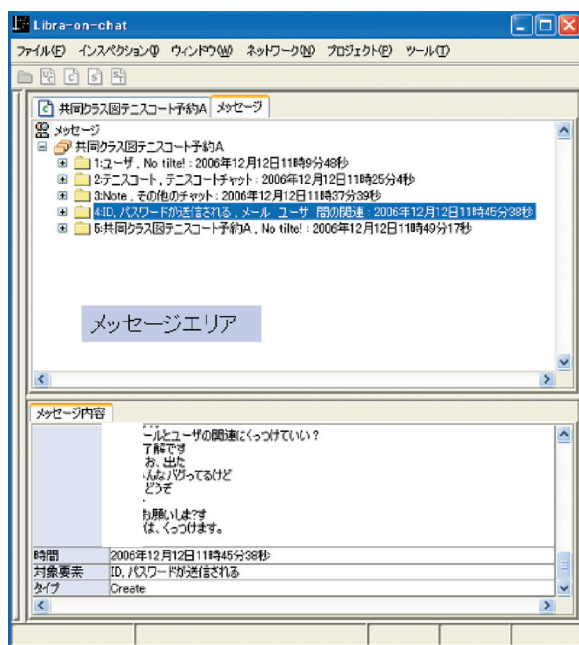


図 6 会話内容の時系列表示

Fig. 6 Historical view of conversations.

程を最初から再現できる（本システムでは「メッセージエリア」と呼ぶ）．

4. 評価実験

本システムの有効性を検証するために，以下の 2 つの実験を行った．実験 1 は，モデリング時におけるダイアグラムと会話内容の関連付けの有効性を明らかにすることを目的とした実験である．実験 2 は，見直し時におけるダイアグラムと会話内容の関連付けの有効性ならびに会話内容の時系列表示機能の有効性を明らかにすることを目的とした実験である．以下，実験方法ならびに結果について述べる．

4.1 実験方法

4.1.1 実験 1

3名の被験者からなるグループを 2 つ構成した（それぞれグループ A，グループ B と呼ぶ）．被験者は UML ダイアグラムを作成してソフトウェアの分析，設計を行った経験を有する大学院生と大学生である．各グループは本システムと，RD-UML のようなコミュニケーションツールと UML ダイアグラム作成機能が統合されていないシステム（RD-UML は利用できないため，RD-UML に相当する機能，すなわち，ダイアグラムの共同編集と UML 要素に関連付けを行わないチャットを著者らが実装した．以降「擬似 RD-UML システム」と呼ぶ）の両方を用いて共同でモデリングを行う実験を行った．実験ではテキストで書かれた課題文に基づき，システム分析段階のクラス図作成をタスクとして与えた．モデリング終了後，被験者全員にアンケート調査を行った．実験計画を表 1 に示す．1 回目，2 回目とも以下の手順で行った．

- (i) キックオフ会議（対面）：実験手順や課題の説明，システムの操作練習を行った（約 30 分）．
- (ii) 個人によるモデリング：キックオフ会議の後各被験者が課題文に対して各自クラス図

表 1 実験 1 の実験計画

Table 1 Experiment plan for the experiment 1.

	1 回目	2 回目
課題	ドリルシステム	テニスコート予約システム
グループ A	擬似 RD-UML	Libra-on-Chat
グループ B	Libra-on-Chat	擬似 RD-UML

を作成する(約1時間)。このときグループメンバとのコミュニケーションは行わないよう指示した。

(iii) 分散同期型の会議：個人によるクラス図作成終了後、割り当られたシステム(Libra-on-Chat もしくは擬似 RD-UML システム)を用いて、グループによる分散同期型の会議を行い、グループとしてのクラス図を完成させる(会議の所要時間は約1時間)。

(iv) 分散同期型の会議終了後、被験者にアンケートの回答を要請した。アンケート項目として「UML チャット(モデル要素に関連付けたチャット)は役に立ったか?」をたずねた。

項目は4段階評価(4: 思う, 3: やや思う, 2: あまり思わない, 1: 思わない)とその理由を自由に記述できるフォームを提供した。

4.1.2 実験 2

実験1の1カ月後に、実験2を実施した。実験2では、実験1で使用した「ドリルシステム」の仕様書に2,3の機能を追加した仕様書、実験1の1回目でグループBがLibra-on-Chatを用いて作成したクラス図ならびにシステムに蓄積された会話内容を提供し、追加された機能を含むクラス図を完成させるよう被験者に指示した。機能を追加した仕様書を提供したのはソフトウェア開発の特徴の1つである仕様変更の場面を想定しているからである。この変更を契機として、ダイアグラムの見直しにおいて本システムがどのように有効に機能するかを調べるためである。実験2では実験1に関わっていたメンバ1名を、実験1に加わっていなかったメンバに入れ替えた3名の被験者により実施した。アンケートとして以下の項目をたずねた。

- 見直し時においてモデル要素と会話内容が関連付けられていることの有効性
- 会話内容の時系列表示機能の有効性

各項目は4段階評価(4: 思う, 3: やや思う, 2: あまり思わない, 1: 思わない)とその理由を自由に記述できるフォームを提供した。

4.2 実験結果

4.2.1 実験1の結果

実験の結果として、作成されたダイアグラムの規模、アンケート結果ならびに会話内容の分析を示す。

(1) 作成されたモデル図の結果

表2に各グループにより作成されたダイアグラムの規模を示す。

(2) アンケート結果

表2 作成された成果物の規模
Table 2 Size of created artifacts.

課題	1回目		2回目	
		ドリルシステム		テニスコート予約システム
グループA	クラス数	8	クラス数	4
	関連数	4	関連数	3
	属性数	14	属性数	11
グループB	クラス数	9	クラス数	4
	関連数	10	関連数	4
	属性数	25	属性数	12

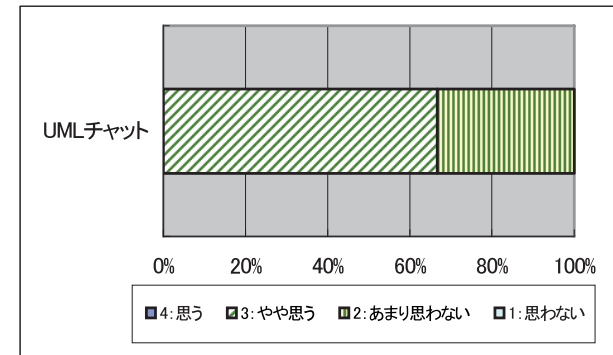


図7 実験1のアンケート結果

Fig. 7 Result of questionnaire of the first experiment.

実験1に参加した6名から得たUMLチャットの有効性に関するアンケート結果を図7に示す。UMLチャットは半数以上が有効であったと回答している。

UMLチャットが有効な理由は以下のとおりであった。

- モデル要素1つ1つに対して議論ができるので議論に集中しやすい。
- 今回のクラス図は規模がそれほど多くないが、クラス図の規模が大きくなったときによ

り効果があるであろう(2名)。

一方、UML チャットの問題点として以下の指摘がなされた。

- 複数のクラスに対する議論が同時に行われ混乱した。
- どこで議論が行われているのかが分かりにくい。

(3) 会話内容を観察した結果

モデリング作業の会話内容を観察することにより、本システム利用による以下の特徴が分かった。

本システム利用時に、ある被験者が、あるモデル要素作成者に対して、多重度、属性について質問するときに、「この(1)って何? 多重度?」、「このIDはStringでいいか?」のように聞いていた。一方、比較対象のシステム利用時には「ユーザクラスの属性にメールアドレスは必要だ」、「会員クラスの性別の属性はintでいいか?」のように議論を行っていた。このことから、本システムを利用する場合、議論において要素を指定するための情報として、クラスや属性名を記述することなく、指示代名詞で要素を指定することができ、対象要素指定のための入力を省くことができることが分かった。

4.2.2 実験2の結果

実験2に参加した3名から得たアンケート結果を図8に示す。特にモデル要素と会話内

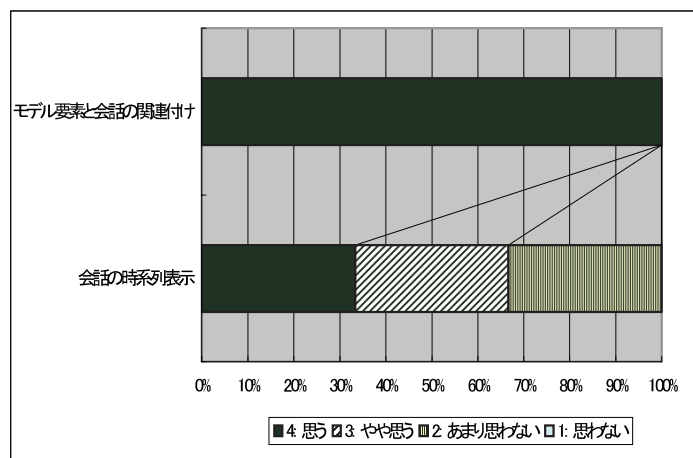


図8 実験2のアンケート結果

Fig. 8 Result of questionnaire of the second experiment.

容の関連付けについては被験者全員が役に立ったと回答している。次に、それぞれのアンケート項目に対する回答理由を示す。

モデル要素と会話内容の関連付け機能の効果

モデル要素と会話内容の関連付けが有効な理由は以下のとおりであった。

- モデル要素に対する会話内容に、その設計に至った根拠が書かれていたため、それを把握したうえで、はっきり不具合であると判定できた。
- 対象のモデル要素に関する議論内容を見ることができるのでよいと思う。特に、自分が初めて(見る)クラス図に対しては大変参考になった。

会話内容の時系列表示機能の効果

会話内容の時系列表示機能について以下のような意見が寄せられた。

- 共同クラス図のアノテーションに番号がふられているので、優先的に作られた(クラスが作られた順番を意味する)クラスが分かるので、より役に立つ(2名)。
- 今回の例ではなぜこのクラス(メールクラス)があるのか知りたいところにチャットがなかったため、残念。

4.3 議論

本研究の目的としてあげた以下の2項目に対して、得られた結果に基づき議論を行う。

(1) モデリング時におけるモデル要素と会話内容との関連付けの有効性

アンケートの結果、実験1の結果からUMLチャットに対して半数以上の被験者がその有効性を認めた。具体的には、モデル要素単位に議論を集中しやすいという利点があげられた。一方、「UMLチャットがより効果を発揮するのはダイアグラムの規模が大きくなる場合であろう」という感想も寄せられており、より大規模なダイアグラム作成での実証実験が必要であることも明らかになった。このとき、作成されたクラス数が10個程度という小規模の今回の実験でさえも、どこで議論がなされているか分からないという指摘があった。この問題はダイアグラムの規模が大きくなればなるほど顕在化すると予想されるので、議論が行われているダイアグラム上の場所や誰が発言しているのかが見て分かるようなアウェアネスに対する強化を行ったうえで大規模ダイアグラム作成に関する実験を行う必要がある。

(2) 見直し時におけるモデル要素と会話内容との関連付けならびに会話内容の時系列表示の有効性

4.2.2項の見直し時におけるモデル要素と会話内容との関連付けによる有効性に関するコメントから、モデル要素を単位とした根拠が残されていると、それは有効活用できることが分かった。また、会話内容の時系列表示機能に関するコメントから、本研究で開発した議論

の時系列一覧表示は議論内容とそれに対する成果物を関連付けて把握するために有益であることを示している。実際のソフトウェア開発において仕様変更や設計変更等の際に設計根拠等の確認は必須作業となるが、それらは議事録やメール等に散在して蓄積されていることが多く、それを当該成果物と関連付けてトレースするにはかなりの労力が必要となる。

本研究が提案したように、モデル要素と会話内容とが関連付けられていれば、トレースのための労力削減やその品質が改善されることが期待できる。その一方、必要な根拠が残されていないという問題も指摘されており、会話内容を記録する際、根拠が捕捉できるように、議論を誘導する必要があることも分かった。

5. おわりに

本論文では、UML ダイアグラムを対象とした、分散共同モデリング支援システムについて述べた。

実験から、モデリング時においてモデル要素と会話内容を関連付けることにより、対象要素を直接指定することが可能となり、対象要素を識別するための入力負荷を減らすことができることが分かった。また、モデル要素に注目しやすく、設計根拠が残るためレビューや見直し時に、UML ダイアグラムの設計方針、モデル要素の設計意図の把握が容易である、という本システムの有効性が得られた。

一方、今回は実験規模が小さく、有益性を実感できない被験者もいた。また、「他のメンバは編集中心なのか、コメントを考えているのか、コメントを打っているのかに関するウェアナース情報がほしい」、「チャット画面がいっぱいになる場合、今どこで話が進んでいるのが分かりにくい」という問題点の指摘もあった。ウェアナースの重要性を指摘したコメントである。また、現在の実装では会話内容は時系列で管理しているが、モデル要素の変化のログを保持していない。会話内容と関連付けてダイアグラムの進化を管理できるようにする必要があり、今後はこれら問題点を改善し、大規模な適用実験を行う予定である。さらに、設計根拠を捕捉するための議論支援についても検討を進めていきたい。

参 考 文 献

- 1) Boulila, N.: Supporting Distributed Software Development with RD-UML, *Proc. Informatiktage 2002*, pp.107-111 (2002).
- 2) Churchill, E.F., Trevor, J., Bly, S., Nelson, L. and Cubranic, D.: Anchored Conversations: Chatting in the Context of a Document, *Proc. 2000 ACM Conference on Human Factors in Computing Systems (CHI2000)*, pp.454-461, ACM Press (2000).

- 3) オージス総研: Elapiz. <http://www.ogis-ri.co.jp/otc/products/Elapiz/>
- 4) Moran, T.P. and Carroll, J.M.: *Design Rationale: Concepts, Techniques and Use*, Lawrence Erlbaum Assoc Inc. (2004).
- 5) 宗森 純, 上坂大輔, タイミンチー, 吉野 孝: 位置情報を用いた汎用双方向ガイドシステム xExplorer の開発と適用, *情報処理学会論文誌*, Vol.47, No.1, pp.28-40 (2006).
- 6) 西田健志, 五十嵐健夫: Lock-on-Chat: 複数の話題に分散した会話を促進するチャットシステム, *コンピュータソフトウェア*, Vol.23, No.4, pp.69-75, 日本ソフトウェア科学会 (2006).
- 7) 角 康之, 伊藤 惇, 西田豊明: PhotoChat: 写真と書き込みの共有によるコミュニケーション支援システム, *情報処理学会論文誌*, Vol.49, No.6, pp.1993-2003 (2008).
- 8) UML Resource Page. <http://www.UML.org>
- 9) Weng, C. and Gennari, J.H.: Asynchronous Collaborative Writing through Annotations, *Proc. Computer-Supported Cooperative Work (CSCW2004)*, pp.578-581, ACM Press (2004).
- 10) Sun Developer Network, Java Shared Data Toolkit (JSDT). <http://java.sun.com/products/java-media/jsdt/>

(平成 20 年 4 月 3 日受付)

(平成 20 年 10 月 7 日採録)



徐 冬梅

2002 年中国牡丹江師範学院計算機科学技術学部計算機教育専攻卒業。2007 年東京学芸大学大学院教育学研究科総合教育開発専攻修士課程修了。同年 NEC ソフト株式会社入社。在学中、協調ソフトウェア開発支援に関する研究に従事。



黒木 純

2005 年東京学芸大学教育学部情報教育専攻卒業。同年株式会社 NTT データイントラマート入社。在学中、協調ソフトウェア開発支援に関する研究に従事。



大瓶 佳秀

2003年東京学芸大学教育学部教育情報科学専攻卒業。2005年東京学芸大学大学院教育学研究科数学情報科学専攻修士課程修了。オブジェクト指向ソフトウェア開発，ソフトウェアインスペクションに興味を持つ。



櫛山 淳雄（正会員）

1985年早稲田大学理工学部工業経営学科卒業。1987年電気通信大学大学院計算機科学専攻修士課程修了。博士（工学）。日本電気株式会社勤務を経て，1999年東京学芸大学助教授。2007年同准教授。1999～2003年電子情報通信学会和文論文誌D編集委員，2003～2007年本学会論文誌編集委員。協調ソフトウェア開発支援，ナレッジマネジメント，協調学習等に関する研究に従事。IEEE Computer Society，ACM，電子情報通信学会，人工知能学会，教育システム情報学会各会員。