

モデル図を対象としたインスペクション 支援システムの構築

大瓶 佳秀¹ 櫛山 淳雄²

本研究は、ソフトウェア開発の上流工程における成果物であるモデル図を対象としたインスペクションを支援するシステムの構築を目的とする。従来のシステムでは、モデル図を対象としたインスペクションの支援において、(1)伝達性の欠如、(2)複数視点への考慮の欠如、という2つの問題点があった。本研究では、それらの問題点を解決するためUMLで定義されているノートを利用したインスペクション支援手法を提案する。また、手法を実現するインスペクション支援システムの設計・実装を行った。本論文では、構築したインスペクション支援システムの詳細について述べる。

Implementation of an Inspection Process Support System for Model Diagrams

Yoshihide Ohgame and Atsuo Hazeyama

The goal of this study is to construct an inspection process support system for model diagrams. Existing inspection process support systems have the following two drawbacks for model diagrams inspection, (1) lack of ability of conveyance, (2) lack of multi viewpoints support. To solve these drawbacks, we proposed a method for supporting model diagrams inspection, which uses “note” defined in UML (Unified Modeling Language). We developed an inspection process support system, which implemented the method.

¹ 東京学芸大学大学院, Graduate School of Education, Tokyo Gakugei University

² 東京学芸大学大学院, Graduate School of Education, Tokyo Gakugei University

1. はじめに

本研究では、ソフトウェア開発の上流工程における成果物であるモデル図を対象としたインスペクション[2]を支援するシステムの提案を行う。

近年、オブジェクト指向によるソフトウェア開発が主流になりつつある。オブジェクト指向による開発では、UML[8]等により記述されたモデル図が分析・設計工程における成果物として作成される。それらのモデル図は実装工程において分析・設計図として用いられる。従って、モデル図に欠陥が存在した場合、実装工程に大きな影響を与えるため、上流工程の段階でモデル図を検証し、欠陥の検出およびその修正を行うことが重要な課題となる。その課題を解決するための有効な評価技法の1つとしてインスペクションがあり、その一連のプロセスを支援するシステムが研究されている[3][4][6]。それらのシステムは、下流工程の成果物であるプログラムコードを対象としており、上流工程の成果物であるモデル図を対象としたインスペクションを主として支援するシステムはほとんど存在していない[5]。

そこで本研究では、従来のインスペクション支援システムによりモデル図を対象としたインスペクションを行う場合、(1)インスペクタが検出したモデル図内の欠陥箇所をモデル図の著者に対して正確に伝達することを支援する機能、(2)複数種類のモデル図のモデル図間整合性検証を支援する機能、という2つの機能が欠如しているという問題点に着目した。そしてそれらを解決するインスペクション支援システムの設計を行った[7]。本論文では、設計に基づき実装を行ったインスペクション支援システムの詳細について述べる。以下、2章では、従来のインスペクション支援システムの問題点について述べ、3章でその問題点を解決するために構築したインスペクション支援システムについて述べる。4章でまとめと今後の課題を述べる。

2. 従来の支援システムの問題点

2.1. インスペクション

インスペクションは、成果物の著者、成果物の検証を行うインスペクタ、インスペクションの運営・管理を行うモデレータを参加者として実施される。また、以下の～ような主要プロセスにより進行される。

成果物の著者がインスペクションを要求する
モデレータにより選出されたインスペクタが
成果物の検査を行う

インスペクタが検査により検出した成果物の
欠陥を成果物の著者に指摘する

成果物の著者が成果物の修正を行う

モデレータが成果物の修正の確認を行い、イン
スペクションが終了となる

インスペクション支援システムは、以上のすべてのプロセスをシステム上で実行できるよう支援することが求められる。

2.2. 従来の支援システムの問題点

従来のインスペクション支援システムは、インスペクションの対象成果物としてプログラムコードを想定しているものがほとんどである。従って、それらのシステム上でモデル図を対象とするインスペクションを行う場合に大きく2つの問題点がある。

(1) 伝達性の欠如

インスペクションの主な目的の1つであり、システムが支援しなければならない活動として、インスペクタが検出した成果物の欠陥を成果物の著者に正確に伝達することが挙げられる。それを達成するためには、以下の2点が支援システムを介して正確に伝達される必要がある。

欠陥の箇所

成果物に含まれる欠陥がどこにあるのかということ

欠陥の内容

成果物に含まれる欠陥はどのようなものであるのかということ

プログラムコードを支援対象とした従来のシステムでは、インスペクタがシステム上に表示されたブ

ログラムに対し、行番号を指定することで欠陥の箇所を、指定した行番号に対する欠陥の指摘コメントを自然言語で記述することで欠陥の内容を伝達することができる。成果物の著者は、行番号に対応して指摘されたコメントを確認することができるため、欠陥の箇所と内容を容易に理解することができる。しかし、モデル図はプログラムコードのようなテキスト文書ではないため、欠陥の箇所を行番号で指定することはできない。欠陥の箇所と内容をすべて自然言語で記述することは可能であるが、その場合、インスペクタの検査効率の低下、成果物著者の読解労力の増加、また、それらを要因とする見解の相違などが生じる危険性がある。従って、従来の支援システムにはモデル図を扱う場合に伝達性が欠如しているといえる。

(2) 複数視点の考慮の欠如

ソフトウェア開発におけるモデル図にはソフトウェアの機能的側面、構造的側面、振る舞いの側面などの複数の視点に対応した複数の種類が存在する[1]。従って、インスペクションの対象となるモデル図が複数種類存在する場合、インスペクタは特定の視点に対応しているモデル図の単体としての検証だけでなく他の視点に対応しているモデル図との整合性という観点からの検証を行う必要がある。また、成果物の著者はモデル図の単体としての検証結果の確認だけでなく、複数のモデル図を横断した検証結果を確認する必要がある。しかし、従来の支援システムでは、インスペクタが複数のモデル図間の整合性を検証することや成果物の著者が複数のモデル図を横断した検証結果を確認するといったモデル図の特質を考慮した支援を行っていない。

3. インスペクション支援システム

本章では、前述した従来のインスペクション支援システムの問題点を解決するために構築した支援システムについて述べる。

3.1. システム概要

本研究で提案するインスペクション支援システム

のシステムアーキテクチャを図 1 に示す。

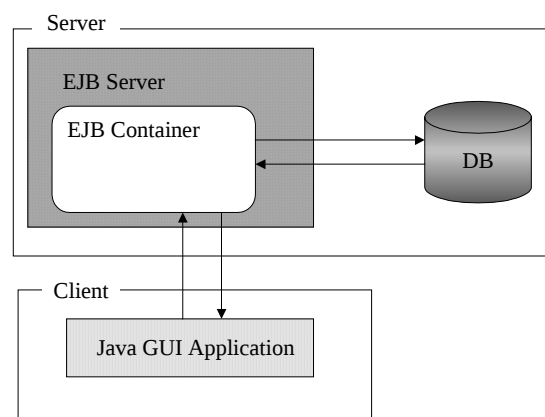


図 1 システムアーキテクチャ

システムは、Java により実装されたクライアント側アプリケーションとサーバ側アプリケーションから構成される。クライアント側は、Java の Swing コンポーネントを使用した GUI アプリケーションとなっている。サーバ側アプリケーションは、EJB コンテナを包含した EJB サーバ上で起動する。なお、サーバ側の DB には PostgreSQL を使用している。

3.2. ノートを利用したモデル図検査支援

本研究では、(1) の問題点を解決するための手段として UML で定義されているノートに着目した。ノートの使用例を図 2 に示す。

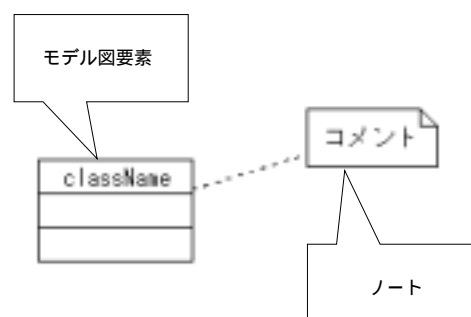


図 2 ノートの使用例

ノートとは、モデル図上にコメントを表現するた

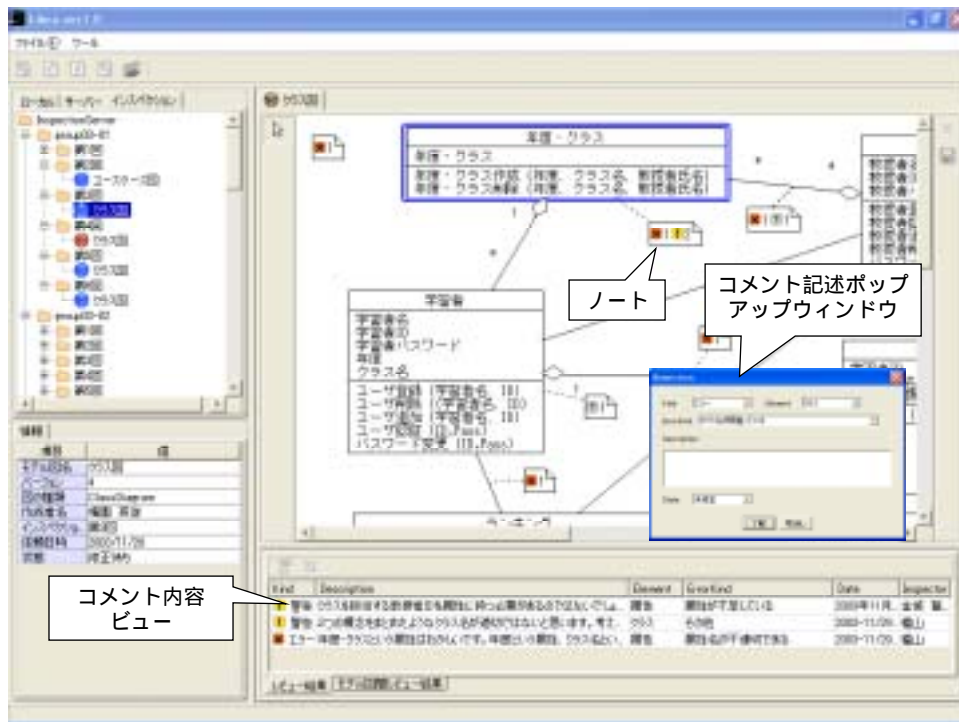


図 3 画面例：ノートを利用したコメント記述

めのグラフィカルなシンボルである。ノートはコメントの対象となる任意の要素に対して指定することができ、モデル図の要素に破線で付加することで表現される。本研究ではインスペクションにおいてインスペクタがモデル図の検査により検出した欠陥を指摘する場合、ノートを利用することで（１）の問題点に対処した。前述したように、欠陥の伝達で重要となるのは、欠陥の箇所と内容である。ノートを利用することで、欠陥の箇所をノートの付加されたモデル図要素、欠陥の内容をノートに記述されたコメントという形式で伝達することができる。これは、プログラムコードの行番号に対してコメントすることに相当する効果が期待できる。しかし、ノートを利用する場合にも問題がある。ノートに記述されるコメントの内容が長い場合やモデル図内に追加されるノートの数が多数になる場合、モデル図が持つ意味的な情報とは無関係な情報が増えてしまうため、本来モデル図が持つグラフィカルな特性を損なう可能性がある。そこで、本研究で提案する支援システムでは、ノートに欠陥の箇所と内容に分離した２つ

のビューで表示することでその問題を解決した。図３に本システムの画面例を示す。インスペクタは、本システム上に表示されたモデル図の欠陥を検出した場合、欠陥のある要素を選択し、表示されたポップアップウィンドウにコメントの記述項目の内容を記述する。コメントの記述項目を以下に示す。

- ・コメントの種類（選択式）

表１を参照

- ・指摘コメントの記述（記述式）

モデル図の欠陥についてのコメント

- ・要素名（選択式）

欠陥のあるモデル図要素の種類

- ・欠陥の種類（選択式）

事前に定義された各モデル図要素に対応する欠陥の種類

- ・日付（自動取得）

コメントが記述された日付

- ・インスペクタ名（自動取得）

コメントを記述したインスペクタの名前

インスペクタがコメント記述を完了すると、シス

テムは対象となる要素に破線で付加されたノートモデル図内に自動生成する。また、欠陥の対象がモデル図の要素ではなく、モデル図の構造に対するものなどの抽象的なものである場合は、要素を選択せずにコメントを記述することができる。その場合、破線の付加されていないノートが自動生成される。ノート内にはコメントの種類に対応したアイコンと種類ごとのコメント総数が表示される。アイコンの種類と意味を表1に示す。

表 1 アイコンと対応するコメントの種類

アイコン	コメントの種類	意味
	エラー	修正が必要な欠陥がある箇所
	警告	欠陥ではないが修正したほうが望ましい箇所
	質問	成果物の著者へ質問がある箇所
	モデル間不整合	他のモデル図との不整合がある箇所

コメントの内容はモデル図の下部のコメント内容ビューに表示される。インスペクタによる検査終了後、成果物の著者がノートの付加された要素を選択することで、コメント内容ビューに各ノートに対応するコメントの内容が表示され、検査結果を確認することができる。このような手法により、欠陥の箇所をノートの付加された要素(またはモデル図自体)、欠陥の内容をコメント内容ビューに表示されるコメント記述という形式でユーザに伝達することができる。さらに、ノート内に表示される情報はアイコンにより省略されているため、モデル図に意味的に効果のない情報を追加することを軽減することができる。

3.3. 複数視点への考慮

本研究は、(2)の問題点についても(1)の問題点と同様にノートを利用して解決する手法を用いた。複数のモデル図の整合性検証を行う場合の本システムの画面例を図4に示す。インスペクタは、タブ型のウィンドウに表示された複数のモデル図をタブの

切り替えにより参照することができる。インスペクタがモデル図間の不整合を検出した場合、不整合のあるモデル図A内の要素を選択したのち、不整合の対象となるモデル図B内の要素を選択し、表示されたポップアップウィンドウの項目を記述することで複数のモデル図を横断してコメントを行うことができる。インスペクタがコメント記述を完了すると、3.1 同様に各モデル図内にノートが自動生成される。本システムの特徴は、モデル図間に記述されたコメントが、自動的に各モデル図に反映されるため、インスペクタがモデル間にコメントを行う場合、各モデル図に対して個々にコメントを記述する必要がないということである。さらに、モデル図Aに一度記述されたコメントを後に改訂した場合、不整合の対象となるモデル図Bのコメントもシステムが自動的に同様の改訂を行うようになっている。この機能により、複数視点に対応するモデル図の整合性検証を行う場合、インスペクタの作業効率の向上が期待できる。また、成果物の著者が、モデル図間の整合性検証の結果を確認する場合、モデル図Aをシステム上に表示するように選択するとシステムはモデル図Aに関連してコメントされているモデル図Bを自動的に同時に表示する。さらに、モデル図間を横断して記述されたコメントは、モデル図間の不整合を意味するアイコンによりモデル図内にノートとして表示され、コメント内容ビューには対応するモデル図要素とコメント内容が表示される。従って、成果物の著者は、視覚的かつ直感的にモデル図間の不整合箇所を確認することができる。

3.4. 非同期分散の支援

本システムは、非同期分散環境下でのインスペクションを支援対象としている。インスペクタによるモデル図の検査はかなりの時間を要する作業である。検査の間、サーバ側に保持されるモデル図のデータをコメント記述のたびに随時更新する方法の場合、ユーザが常時ネットワークに接続されていなければならないという制約や複数のインスペクタが同時ア

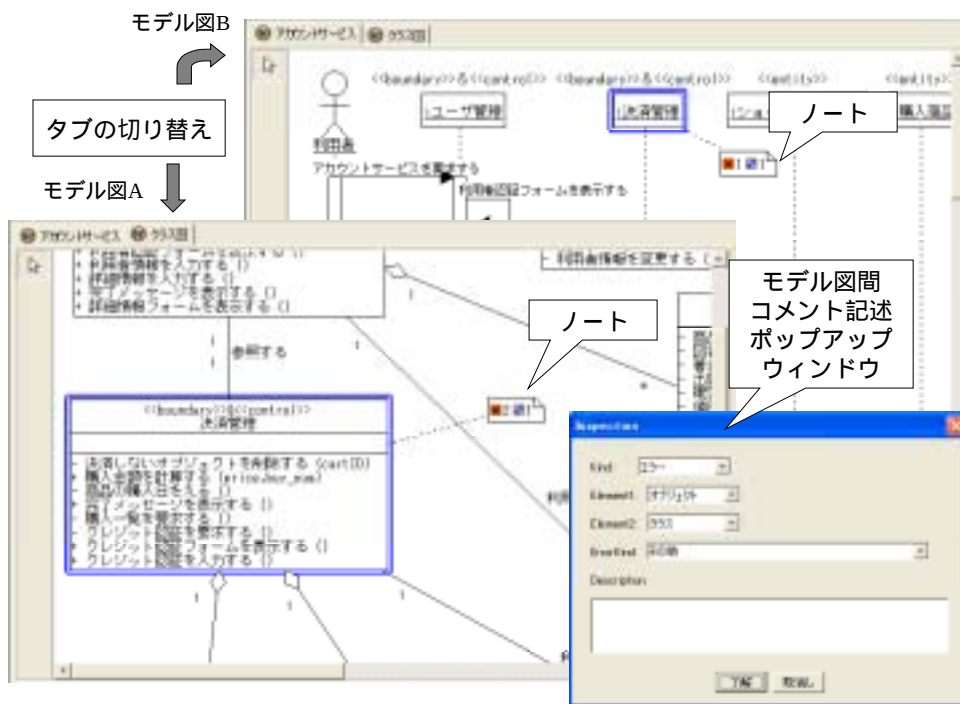


図 4 画面例：複数のモデル図間の整合性検証

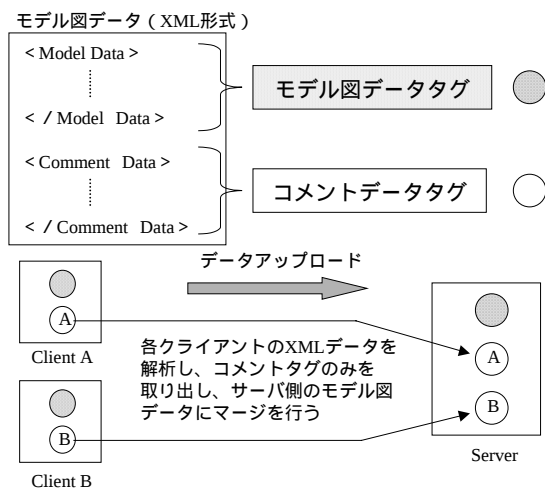


図 5 モデル図のマージ

アクセスした場合の排他制御によるサーバ側の負担などの問題がある。従って、本システムでは、インスペクタがモデル図の検査を行う場合には、一旦最新のモデル図をクライアント側にダウンロードし、モデル図の検査を行った後、コメント情報の付加されたモデル図をアップロードするという方法を採用した。しかし、この方法においてもモデル図の更新に問題がある。複数のインスペクタが同一情報のモデ

ル図をダウンロードして各々コメントを行った場合、情報の異なる複数のモデル図が各クライアント側で作成されることになる。それらのモデル図をサーバ側にアップロードすると不整合が生じてしまう。そこで、本研究ではコメントの付加されたモデル図データをマージする機能を用意した。図5にモデル図のマージを示す。本システム上で扱うモデル図はXML形式で保持される。モデル図に対してコメント情報が付加されるとモデル図の要素タグ以外にコメントのタグが追加される。コメントのタグ内にはインスペクタのIDの情報が含まれる。本システムでは、モデル図がサーバ側にアップロードされた場合、XMLデータの解析を行い、ID情報からアップロードしたインスペクタのコメントのタグをサーバ側にあるモデル図のXMLデータに対し追加することでXMLデータのマージを行う。これによりサーバ側にあるモデル図はすべてのインスペクタのコメントが反映された状態で管理することができる。

3.5. UML エディタ機能

インスペクションは、他のレビュー技法とは異なる

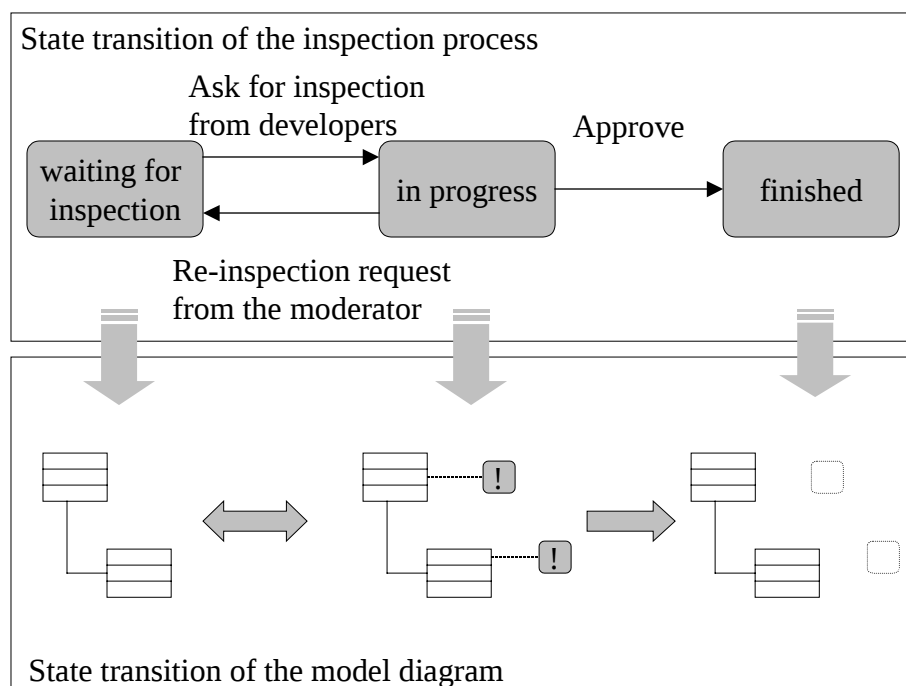


図 6 インспекションの状態遷移とモデル図の状態

り成果物の検査だけでなく成果物の修正という作業を含む。従って、インспекション支援システムは、システム上で成果物を改訂できることが必要である。そこで、本研究では、システム上で UML のモデル図を編集できる機能を用意した。また、システム上で作成したモデル図は、サーバ側でバージョン管理して保存することができる。この UML エディタ機能により、モデル図の作成、モデル図の検査、モデル図の欠陥の伝達、モデル図の修正というインспекションのすべてのプロセスをシステム上で行うことができる。

3.6. インспекションの状態管理

本システムの特徴の 1 つとして、モデル図の状態を利用したインспекションの状態管理がある。図 6 に本システムにおけるインспекションの状態遷移とインспекションの各状態におけるモデル図の状態を示す。まず、成果物の著者（開発者）がモデレータにインспекションを依頼した時点では、モデル図は初期状態となる。次に、モデレータにより選出されたインスペクタがモデル図の検査を行い欠

陥の検出を行った状態では、モデル図は前述した手法によりアイコンを表示したノートが追加された状態となる。検査終了後、成果物の著者が欠陥の修正を行った後、モデレータが修正後のモデル図と修正前のモデル図を比較し、欠陥の修正の確認を行う。本システムでは、インスペクタが記述した各コメントの状態（"修正前"、"修正済み"）をモデレータが変更できるようになっている。モデレータが修正の行われた欠陥に対するコメントの状態を"修正前"から"修正済み"に変更すると、モデル図要素に付加されている該当のノートモデル図上からシステムが自動消去する。従って、修正の確認が行われるにつれ、モデル図内のアイコンを表示しているノートが消える。モデレータはモデル図内のアイコンの表示されたノートの有無により修正の確認の終了状況が明示的に認識できるため、確認漏れを防ぐことができる。モデレータによるすべての欠陥の確認が終了した時点で、モデル図にアイコンを表示したノートが残っている場合、欠陥の修正が不足していることを意味するため、成果物の著者に対しモデル図の修

正を再要求する。再度修正を行う成果物の著者は、残ったノートを確認することで、修正の不足した箇所を容易に認識することができる。一方、モデレータによる欠陥の確認終了後、修正アイコンを表示したすべてのノートが消えた場合、すべての欠陥の修正が確認されたことになりインスペクション終了となる。このように、本システムでは、モデル図に対する欠陥の指摘コメントを表現するノートの追加や削除によりインスペクションの状態を管理することができる。また、ノートの有無により、インスペクタの欠陥検出作業、モデレータの修正確認作業、成果物著者の修正作業を視覚的かつ直感的に行うことを可能とする。

4. まとめと今後の課題

本論文では、モデル図を対象とするインスペクション支援システムの構築について述べた。従来の支援システムでは、モデル図を対象としたインスペクションの支援において、“伝達性の欠如”、“複数視点の考慮の欠如”という2つの問題点があった。今回構築したシステムでは、システム上に表示したモデル図に対し、UML で定義されているノートを利用してインスペクタが指摘コメントを記述することにより、ユーザに対して伝達性を提供することを可能とした。また、ノートを利用した指摘コメントを複数のモデル図を横断して記述することにより、複数種類のモデル図間の整合性検証を支援した。

今後の課題としては、システムの有効性の評価があげられる。本システムを、東京学芸大学のシステム開発演習授業のインスペクションに適用した。インスペクションは、授業の教授者1名をモデレータ、大学院生2名および学部4年生1名をインスペクタ、学部3年生23名(計6グループ)を開発者として行われた。今後、適用結果から本システムで提案した支援手法およびシステムの各機能について、有効性の検証を行っていく予定である。

参考文献

- [1] G. Booch, The Unified Modeling Language User Guide, UML ユーザガイド, ピアソンエデュケーション.
- [2] T.Gilb, and D.Graham, Software Inspection, Addison Wesley, 1993.
- [3] L. Harjumaa, and I. Tervonen, A WWW-based Tool for Software Inspection, Proceedings of the 31st Hawaii International Conference on System and Science, Vol. III, pp. 379-388, 1998.
- [4] F. Macdonald, Computer Supported Software Inspection, PhD Thesis, October 1998.
- [5] F. Macdonald, and J. Miller, A Comparison of Computer Support Systems for Software Inspection, Automated Software Engineering 6(3), pp. 291-313, 1999.
- [6] V. Mashayekhi, C. Feulner, and J. Reidl, CAIS: Collaborative Asynchronous Inspection of Software, Proceedings of the second ACM SIGSOFT Symposium on the Foundations of Software Engineering, 1994.
- [7] 大瓶佳秀, 樋山淳雄, モデル図を対象としたインスペクション支援システムの設計, 日本ソフトウェア科学会第10回ソフトウェア工学の基礎ワークショップ(FOSE'03), 近代科学社, pp.125-128, 2003.
- [8] OMG Unified Modeling Language Specification Version1.4, OMG (Object Management Group).