「ソフトウェア可視化技術」討論レポート

大木幹雄　　2/12/03

１．目的

物理現象を扱う分野では，現象の背後にある因果規則や複雑な作用を直感的に把握しやすくするため，CGやアニメーションを用いた現象の可視化が行われる．たとえば，図１で示す障害物の背後で生じる渦の挙動や密度変化等は代表的な事例である．このような可視化によって，現象のもつ意味や洞察を一層深めることができる．

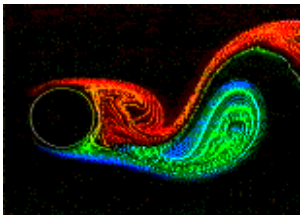

図１　円筒形の障害物の背後で生じる過流

ソフトウェアに関して同様な可視化を試みるとき，物理現象に比較して，「現象の構成要素自体がもつ変化規則」の究明や「構成要素間の関係の変化規則」等の理解が著しく遅れていることに気がつく．実際，ソフトウェアの分類構造や静的な情報量（たとえば，クラス構造やモジュール構造，統計量，各種の分析設計図式等）の可視化は従来から行われてきたが，ソフトウェアのもつダイナミクス（たとえば，バージョンアップがもつ進化の原理や要求仕様が収束するときのプロセス等)に関しては,ほとんど可視化されていない．ソフトウェアの可視化に関しては，いまだプラトンの“what are things made of?“の時代であり，ニュートンの“Why do things move as they do?”の時代にすらいたっていないと言えよう．

さらにソフトウェアのもつ抽象概念の表現方法が物理現象の表現よりも，人間の認知過程に深く依存していることも可視化技術の研究を困難にしている．　これは逆に考えると，人間の認知過程まで立ち入ったソフトウェア工学研究が必要とされている現況において，可視化技術は大きな役割を果たしうることを示している．

そこでワークショップは，図２のような可視化対象および課題モデル図をたたき台として描き，可視化目的と表現空間との対応関係，あるいは可視化目的と認知的・物理的表現空間の選択方法や効果予測等を中心に討論を展開することを予定した．

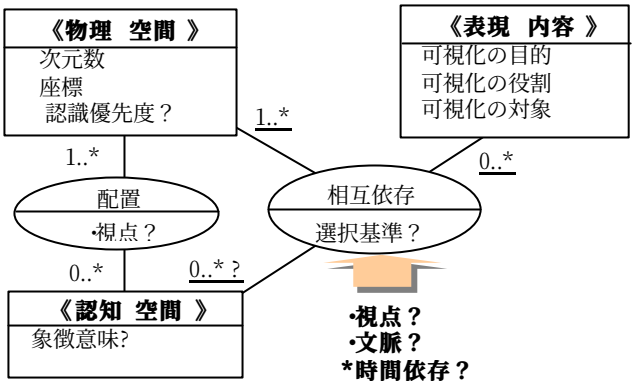

図２　　ソフトウェア可視化技術の課題構造

２．　参加者と討論内容

(1) 参加者

参加者は次のとおりであった（五十音順）．グループは少人数ではあったが，少人数がゆえに精鋭の方々ばかりで，可視化技術がソフトウェアの理解や分析設計・開発に与える効果について，それぞれの立場から活発に意見が出された．

・臼井義美（臼井技術士事務所）
・大木幹雄（日本工業大学）
・神林　靖（日本工業大学）
・黒坂靖子（日本電子計算）
・清水誠介（立命館大学）
・丸山勝久（立命館大学）

(2) 討論内容

【第１日目】

各メンバのポジションペーパにしたがって，可視化に関する研究成果と課題の提起が行われた．以下，提起された課題のみを示す．

① プログラミング初等教育におけるプログラム動作可視化ツールの効果（大木幹雄）

ソフトウェアがもつ特性の状態変化や分析設計内容の動的な変化，プログラムの動作等のソフトウェアがもつ動的側面の可視化が理解に不可欠であるが，その実現は遅れている．

② クラス間関係の簡略によるフレームワークの可視化（清水誠介，丸山勝久）

アプリケーションを構成するクラスとフレームワーククラスとの境界に重要度付けを行う理解手法が必要になる．その際に，より有効な表示方法を考慮しなければならない．これはプロダクトの可視化全体について言える．

③ 分散オブジェクト環境におけるシステムの可視化に関する一考察（臼井義美）

データあるいはオブジェクトの流れにしたがって，ソフトウェアの構成を可視化するとシステムの合成や再利用が容易になる．このような可視化は，人間の思考過程の本質（ものの動きの把握）に根ざしているのではなかろうか．

④ プログラミング環境におけるテキストとグラフィック表現の平衡(神林靖)

可視化において，グラフィカルな表現とテキスト表現の併用が効果を高める上で必要になる．しかしどのような役割分担・目的で使い分けるべきか，定量的な研究や適用基準が明らかになっていない．

⑤ AVS/Express にみるコンポーネント可視化合成環境の課題(黒坂靖子)

可視化ソフトとして世界的に有名な AVS であるが，開発者の最近の関心は，人間の認知過程にまで立ち入った可視化方法に移行している．可視化プログラムがもつ機能の高度化に伴って，可視化コンポーネントの合成空間の次元を 2 次元から 3 次元に拡張する必要が出ている．

【意見交換パーティ】

新たなアイデアやニーズを引き出す上で討論以上に意見交換パーティが重要な意味をもつ．パーティ会場で他の討論グループから，最近の移動体システムのトラブルに関して，原因箇所の特定が困難であり，ソフトウェア品質保証や障害対策の面で可視化技術がどのような役割を果たすのかについて問われた．この出来事は翌日の討論に大きな影響を与えた．

【第２日目】

1 日目の討論を受けて，可視化すべき具体的な対象と優先度から討論を開始した．発散の防止するため，設計仕様の可視化にターゲットを絞り，目的，手段等を考察した．その結果，設計仕様は，開発時のみでなく，「障害発生において原因の特定や影響範囲を特定する」役割の方が大きいとの問題提起がなされた．たとえば，ハード製品やプラント開発では，設計書は開発時より，むしろトラブル発生時に重要な役割をもつ．しかし，ソフトウェアでは，設計書は開発が完了すると無用の長物になる．この違いはどこからくるのであろうか？最近のゼロックスのコピーマシンは,障害発生時に障害箇所と復旧手段を可視化して示してくれる．それを可能としているのは，設計時から障害対策機能の組込であり，人間にとって理解しやすい障害箇所の表示にある．

従来の捉え方では可視化技術は，単なる出力系に関する技術の色彩が強かった．しかし今後は，コピーマ

シン同様，可視化技術は，機能や障害を理解する人間に対する入力系として，システム設計時の重要な制約となろう．その意味で可視化技術は今後のソフトウェア技術のあり方に変更を迫るキー要因になるかもしれない．

3． 成果

認知過程に立ち入った具体的な討論にはいたらなかったが，以下に述べる設計仕様の可視化目的と課題，実現手段等について検討し内容を掘り下げたことは，今後の可視化技術のあり方に一つの方向性を与えた．

●設計仕様の可視化に関して

(1)主な目的

①ユーザ，開発メンバとのコミュニケーション，および理解の促進

②障害発生時の原因，影響範囲の特定

(2)課題

①理解しやすい内容の表現形式とは？

図式表現については，すでに種々の方式が実用に供されているが，クラス図と OCL(オブジェクト制約言語)に見られるような図式とテキストの役割分担の境界線はどこに置くべきか？

②原因の特定に役立つ設計仕様のもつ「意図」を理解にするには？

意図をどのように表現するか？　意図を理解するためには，通常，設計仕様が決定されるまでの過程とその履歴(議事録)が必要になるが，どのように表現したら，理解が容易になるか（たとえば，変更履歴の木構造表現？）

(3) 具体的な理想像

ソフトウェアに障害が発生すると，その原因箇所と対処方法を理解しやすい形式で可視化してくれること．そのためには，意図も含めて，たとえば以下の問題をどう解決すべきか？

①障害発生検出と対策の設計手法と設計図式とは？

②プログラム言語の設計との対応付け方法は？

③設計履歴に含まれる意図の理解手順，意図を理解した対応箇所の特定方法と可視化方法は？

4． 今後の方向

引き続きソフトウェア可視化技術を鳥瞰し課題を体系的に捉えると同時に，具体的な可視化対象を選択して議論を今後も地道に重ねる必要がある．今回のワークショップは障害発生と，その原因箇所と対処方法を可視化する方法について討論を重ねたが，今後も可視化の効用について具体的な応用例をとりあげながら，議論を重ねてゆく必要がある．

関連する国際会議，学会

[1]VMSE(Visual/Multimedia Software Engineering)
http://www.cs.dal.ca/HCC03/VMSE

[2]可視化情報学会(VSJ)
http://www.VSJ.OR.JP