

解説

ユーザの視点を取り入れる技術：

システム開発におけるシナリオの役割

郷 健太郎
山梨大学

John M. Carroll
Virginia Tech

今宮 淳美
山梨大学

ユーザに視点を置いた設計プロセスに関するISO13407や、製品の品質保証に関するISO9000のような国際標準、さらに製品の安全性に関する製造物責任法（PL法）の制定により、ユーザの視点をシステム設計に反映させる手法の重要性が増している。そこで、ユーザの視点をシステム設計に取り入れる手法として、「シナリオ」の利用が注目を浴びている。

シナリオとは、システムのユーザが目標を達成するために行う行動と、そこから得られる事象を、時系列に沿って記述したものである。つまり、ユーザがどのような状況で何をするかという、ユーザに視点を置いた又脈とタスクに関する情報を表現している。このような、従来のシステム開発では2次的に利用されてきた情報を、システム開発プロセスにおける主要な成果物として積極的に利用するのが「シナリオに基づく設計・開発」であり、この枠組みが急速に成立しつつある。

本稿では、シナリオに基づく設計・開発の枠組みについて述べる。まず、シナリオとは何か、そしてシナリオがシステム開発工程でどのような役割を持っているかという問い合わせる。さらに、シナリオが積極的に用いられている4つの分野（戦略経営、ヒューマンインターフェース、要求工学、オブジェクト指向分析・設計）について紹介し、それらの関係を説明する。

シナリオが注目されている

システム設計・開発においてユーザの視点を取り入れる技術の重要性が、現在急速に増している。ユーザに視点を置いた設計プロセスに関するISO13407や、製品の品質保証に関するISO9000のような国際標準、さらに製品の安全性に関する製造物責任法（PL法）の制定により、ユーザ側から見て、使いやすく、安全で、作業効率を上げるようなシステムの開発に対して、これまで以上に力を入れる必要が出てきている¹⁾。

そこでユーザの視点をシステム設計に取り入れる手法として、「シナリオ」の利用が注目を浴びている²⁾。

またアカデミックな観点からも、1998年には IEEE Transactions on Software Engineering, Vol.24, No.12 (1998) と Requirements Engineering, Vol.3, No.3-4 (1998) とがシナリオに関する特集号を出し、シナリオに関する研究が脚光を浴びている。この背景には、これまで複数の分野で個別的に行われてきた基礎研究の発達があり、これらが他分野との融合を図り、ソフトウェア開発における具体的・現実的な1つの手法として、

新たな方向性を模索し始めていると考えられる。

本稿では、シナリオに基づくシステム設計の現状、特にシステム設計におけるシナリオの役割について解説する。また、代表的な4分野—戦略経営（Strategic Management）、ヒューマンインターフェース（Human-Computer Interaction, HCI）、要求工学（Requirements Engineering）、オブジェクト指向分析・設計（Object-Oriented Analysis and Design）—での、シナリオの特徴的な使い方を概説する。さらにHCIの分野で提唱されているクレーム分析法について、例を使って説明する。

シナリオとは何か？

シナリオとは、ユーザが目標を達成するために行う行動と、そこから得られる事象を、時系列に沿って記述したものである。対話型のシステムを設計・構築する場合には、何らかの局面で必ずシナリオが用いられるといつても過言ではない。シナリオに基づく設計法とは、シナリオをシステム開発の中核をなす成果物と捉え、積極的

タケシはWindows98のノートパソコンを使いはじめて約1年のユーザである。彼は、ほぼ毎日パソコンを、主に友達と電子メールをやりとりするために使っている。今日も電子メールを3通書いて送った後、パソコンの電源を切ろうと思った。

タケシは、まず【スタート】ボタンをクリックした。表示されたスタートメニューでは、選択項目の下2つ（【タケシのログオフ】と【Windowsの終了】）と他の選択項目との間には、線が引かれて区別されていた。2つのうち、彼は慎重に【Windowsの終了】を選択した。【Windowsの終了】ダイアログボックスが画面中央に表示された。タケシは「次の方法で終了してもよろしいですか？」という質問を一瞬見た後、4つの選択肢をさっと見た。そのうち、2版目の【電源を切れる状態にする】のボタンがいつもどおり選択されているのを確認し、【はい】をクリックした。小さな効果音とともに画面が暗くなり、その後、自動的にパソコンの電源が切れた。

図-1 自然言語によるWindows98終了とパソコン電源切断のシナリオ。
下線は分析対象の相互作用オブジェクトを意味する。

を利用する開発法である。

ヒューマンインターフェースの分野で用いられるシナリオの例を図-1に示す。このシナリオでは、1人のユーザが、Windows98を終了してノートパソコンの電源を切るまでの作業の段階を、順番に表現している。

シナリオの表現

シナリオには次の要素が含まれる：(1) アクター（ユーザ）、(2) アクターとその環境に関する背景情報、(3) アクターの目標、(4) アクションとイベントの列。ただし、各要素がシナリオの文脈から明らかな場合には、そのいくつかを簡潔に記述したり、省略したりすることがある。なお、シナリオ中のアクターは1人だとは限らない。たとえば、多数のユーザがいるグループウェアシステムの場合、アクターが同時に複数存在することがある。

図-1の例では、(1) アクターはタケシであり、(2) 背景として、パソコンの使用経験が1年で、ほぼ毎日利用していること、そして、今日の作業（電子メールの送信）が終了したことが分かる。(3) アクターの目標は、パソコンの電源を切ることであり、その副目標として、Windowsの終了がある。(4)

図-1中の第2段落はアクションとイ

ベントの列を示している。これは、【スタート】ボタンをクリックするアクションから始まり、パソコンの電源が切れるというイベントで終了する。

シナリオ（特にアクションとイベントの列）は、必ずしも図-1のような物語体の文章形式である必要はない。絵を使って表現してもいいし、ビデオのような動画像を用いることもある。システムのストーリーボード（システムの操作画面遷移図）やプロトタイプを組み合わせてもよい。さらに、形式技法として研究・開発されてきた表現形式（たとえば、フローチャート・状態遷移図・シーケンス図に代表される図的表現や、CCS・CSP・Zに代表されるテキスト表現）を利用することも可能である。

表現形式は、シナリオの使用者に対して、柔軟に使い分けたり、組み合せたりすることが必要である。高度な専門知識を持った技術者だけが使うのであれば、個別的な使用目的に適した特殊な人工言語を用いて記述してもよい。しかし、システム設計過程にそのユーザの参加を促すのであれば、誰でも理解できるように、形式性を排除した物語体の文章やビデオなどを用いなければならない。

シナリオの特徴

シナリオは、単に構築されるシステムの要求を文書化するだけでなく、その潜在的ユーザが取り得るシステムとの相互作用を、構築前に明確化するという利点を持つ。さらに、シナリオはソフトウェア開発に携わるすべての人の共通言語となり得る。シナリオはシステムの操作を具体的に描写しているため、読み手がその状況を理解しやすい。つまり、要求を出す側と獲得する側とが、シナリオを通して共通の理解を構築することが可能である。

一方で、シナリオを工学的に扱う場合、次の性質が問題視されることがある：(1) 量と数の両面で、どのくらいシナリオを書いてよいのかはっきりしない、(2) 表現内容に冗長性がある、(3) シナリオの質が作者に大きく影響される。これらの諸性質については、シナリオに関する作成ガイドラインをプロジェクトごとに設定し、また支援ツールを導入すれば、改善できると思われる。しかし、具体的なガイドラインや支援ツールに関しては、現状ではまだ不十分である。

システム開発工程におけるシナリオの役割は、その特徴を、仕様が持つ特徴と比較することで自然に導出される。

シナリオ vs. 仕様

シナリオは仕様ではない：シナリオと仕様とは対極をとる概念である（表-1）。シナリオは、具体的な記述であり、特定の事例や局面を示すものである。また、ユーザが何をどう行うか、という作業に焦点を絞ることで得られる。シナリオは、記述数と量に制限がなく、しかも断片的に対象を捉えている。さらに、通常は非形式的で、大雑把に記述される。ここでは、口語体をとることが多い。さらに、シナリオは心に描いた結果を得るために用いられる²⁾。

これとは対照的に、仕様では、個

別的なアプリケーションに依存しない抽象的な記述を行い、各機能がとるべき一般的な型を記述することが必要である。また、システムがどんな機能を備えるかという、技術に焦点を絞ることで得られる。仕様には、完全で徹底的な記述が必要とされ、しかも形式的で厳正だという性質が求められる。そして仕様は、記述された結果を得るために用いられる。

以上のような特徴から、シナリオと仕様では、ソフトウェア開発工程において果たすべき役割が大きく異なっている。曖昧性を除去した厳密なシステムの記述が必要な場合や、形式的にシステムの検証を行う場合には、形式的な仕様が必要である。しかし、技術者以外の関係者がそのシステムを理解しなければならない場合、非形式的なシナリオを使って、システムの具体的な使用状況を表現しなければならない。つまり、シナリオと仕様とを、お互いに表現しない部分を補完する文書として有機的に利用することが望ましい。

シナリオに基づく 設計・開発

いつ・どこでシナリオを使うのか？
シナリオは、ソフトウェア開発工程のすべての状況で用いられる。実際に任意の工程で、文書化・共通言語・共有知識の役割を果たす。また、各工程に応じて固有の役割がある。これらを表-2に示す。設計中にはまだ存在していないユーザの作業を明確化するというシナリオの特徴から、上流工程ほどその役割に多様性があることが分かる。実際、ワークフローの分析、業務プロセスのモデル化の道具としてシナリオが利用される場合も多い⁴⁾。

誰がシナリオを使うのか？

シナリオはソフトウェア開発にかかる人であれば、誰が書いて、誰が読んでもよい。表現形式として物

シナリオ	仕様
具体的な記述	抽象的な記述
特定のインスタンスに注目	一般的な型に注目
作業から得られる	技術から得られる
開放的、断片的	完全、徹底的
非形式的、大雑把	形式的、厳正
口語体	
心に描いた結果	記述された結果

表-1 シナリオと仕様の比較。
文献2) から抜粋

語体の記述をとれば、工学教育を受けていないユーザや顧客が書いたり読んだりすることも可能である。この意味でシナリオは、ソフトウェア設計へ実際のユーザの積極的な参加を促す材料となり、いわゆる参加型の設計 (Participatory Design) を可能にする。これは、従来行われてきた設計者主導の設計とは、概念がまったく異なるものである。

シナリオの種類

シナリオにはアクターの目標が記述してあるため、その目標に達成できるかどうかで、成功シナリオと失敗（エラー）シナリオに大別できる。

さらに、シナリオが記述する内容について、通常のシナリオと反シナリオ (Anti-scenario) に分けられる。前者では、必要な望ましい目標、アクションやイベントについて記述し、後者は「これはユーザに行って欲しくない」という要求について記述する。

どうやってシナリオを書くのか？

シナリオを作成する作業は、大きく5つのステップに分けられる。

- (1) ユーザクラスの設定：予想されるアクターのクラスを設定する。
- (2) 具体的なユーザの設定：シナリオが対象とするアクターを決める。アクターの背景とシステムの利用状況を明確化するという観点から、その具体性が高い

開発工程	シナリオの役割
要求分析	ブレインストーミングツール ユーザタスク分析ツール プロトタイプ 文書化 共通言語 共有知識
仕様化	仕様補助文書 文書化 共通言語 共有知識
設計	プロトタイプ 文書化 共通言語 共有知識
実装	文書化 共通言語 共有知識
試験	テストシナリオ 文書化 共通言語 共有知識
運用・保守	トレーニング教材 操作マニュアル 保守マニュアル エラー・バグ報告 文書化 共通言語 共有知識

表-2 ソフトウェア開発工程におけるシナリオの役割

ほどよい。

- (3) ユーザの目標の設定：シナリオ中のアクターの到達目標を決定する。
- (4) シナリオの種類の決定：前述したシナリオの種類から選ぶ。
- (5) イベント列の記述：ユーザが起こすアクションと、システムが起こすイベントを、順番に列挙する。

これらの作業は順番に行う必要はない、最終的に完備さえすれば、任意の項目から作業を始めてよい。

シナリオはソフトウェア開発に使われているか？

すでにシナリオは、実際の現場で利用されており、重要な役割を担っている。Weidenhauptのグループは、欧州4カ国にわたって合計15のソフトウェア開発プロジェクトを調査し、その中でシナリオがどう使われているか分析を行った⁷⁾。その結果、シナリオの持つ以下のような性質を明らかにした。

メニュー選択項目：「タケシのログオフ」

- ...現在のユーザ名を表示する。
- ...ユーザに、Windowsを終了せずに使用を終了し、他のユーザが使えるようにする機能を提供する。
- ...しかし、ユーザが単独で使用するシステムの場合、「ログオフ」を必要としないかも知れない。
- ...しかし、単独でシステムを使用するユーザが、間違って「ログオフ」を選択するかも知れない。

ダイアログ選択項目：「電源を切れる状態にする」

- ...ユーザにWindowsの終了と電源の切断を促す。
- ...しかし、ユーザが「電源を切る」とは理解してくれないかも知れない。

図-2 クレーム分析例。図-1のシナリオにおける相互作用オブジェクトに関する分析。

- (1) 抽象モデルの構築に失敗した場合に用いられる。
- (2) 開発者とユーザとのコミュニケーション・相互学習を促す。
- (3) システムのプロトタイプと同時に用いられる。
- (4) 開発プロセスの複雑さを減少させる。
- (5) ステークホルダ^{☆1}間の部分的な合意と一貫性を促す。

特に、彼らは(1)に関して興味深い例を2つ報告している。1つは、業務情報システム開発の大規模プロジェクトに関する事例である。このプロジェクトでは、要求分析チームが、システムが支援する業務手順(ビジネスプロセス)のオブジェクトモデルを構築していた。しかし半年間作業を進めた時点で、モデルがきわめて複雑になり、管理不可能になってしまった。そのため業務プロセスが的確に把握できなくなり、しかも顧客はこれまでに構築したモデルを理解してくれないために、その検証さえできなかつた。そこでプロジェクトはモデルの定義をやめ、シナリオを利用することで、プロジェクトを進めることができるようにになった。具体的には、アプリケーション領域を15のトピックに分割するためにシナリオを利用し、その後、開発チー

ムが各トピックを理解するためにシナリオを利用した。

もう1つの事例は、これよりは小規模の業務情報システム開発プロジェクトである。このプロジェクトでは、当初ペトリネットを使ってビジネスプロセスを定義していたが、120以上のプロセスをモデル化した後で、新たなプロセスを既存のプロセスと関連付けたり、プロセス全体の無矛盾性を保証することができなくなった。そこで、ペトリネットで形式化する作業をやめ、ビジネスプロセスのシナリオを書くことにした結果、27のシナリオで本質的なシステムの利用形態をすべて記述することができた。数が少なくてすんだ理由は、シナリオと比べてペトリネットがきわめて詳細なプロセスを数多く記述していたためだった。

シナリオに基づく設計における技法

ここでは、シナリオに基づく設計における技法の具体例として、クレーム分析法について説明する。

【シナリオの分析法】

クレーム分析とは、タスクシナリオから抽出された相互作用オブジェクトに対し、それらが持つ望ましい性質とそうでない性質とを列挙する作業(または得られたりスト)である。

図-1のシナリオに対するクレーム分析例を図-2に示す。図-1では、下線部の「タケシのログオフ」と「電

源を切れる状態にする」が相互作用オブジェクトである。

各相互作用オブジェクトの下には、そのオブジェクトが持つ望ましい性質とそうでない性質とを列挙する。後者については、「しかし」を文頭につけ、前者との違いを明示している。

「タケシのログオフ」オブジェクトの場合、望ましい性質として、ユーザ名の表示と、他のユーザへの速やかな交代機能が挙げられる。一方で、望ましくない性質として、単一ユーザが使用するときに「ログオフ」機能を使わないかも知れないことと、間違って「ログオフ」を選択してしまう可能性が挙げられる。

クレーム分析法の特徴は、相互作用オブジェクトの持つ望ましい性質とそうでない性質との両方を列挙する点である。これらの性質は、通常トレードオフの関係にあるため、両者を比較・検討することで、設計者や分析者が、そのオブジェクトの性質を公平・正確に把握することができる。

シナリオの研究動向

シナリオに関する研究は、現在までに主として4つの分野で個別的に行われてきた：(1) 戰略経営、(2) ヒューマンインタフェース、(3) 要求工学、(4) オブジェクト指向分析・設計。これらのうち(1)以外は、基本的にソフトウェア開発に関連する分野である。これらの4分野について、シナリオの特徴と代表的な研究・実践を挙げる。

戦略経営

シナリオを系統的に用いるという観点では、戦略経営の分野が4分野の中で最も古い歴史を持つ。1960年代の中期から後期にかけて、Royal Dutch/Shellグループはシナリオを用いてオイルショックに対する危機管理を行った⁶⁾。具体的には、

☆1 ソフトウェア開発に影響を及ぼすさまざまな人々、プロジェクトマネージャ、顧客、エンドユーザを含む。

現在の社会情勢が今後も続くと仮定した場合のシナリオを基本とし、そこから「もし〜ならばどうなるだろう？」という質問（What-if questions）とそれに対する回答を、段階的にシナリオとして作成・分析することで、オイルショックが起こる前に危機対策を行った。

Royal Dutch/Shellのシナリオに基づく計画の成功を受けて、戦略経営の分野ではシナリオに対する研究・実践が行われており、1992年には、Planning Reviewがシナリオに関する特集号を2号にわたって組んでいる。

戦略経営の特徴は、長期にわたるシナリオ（たとえば、5年後の情報通信業界の基盤構造や10年にわたる会社の長期ビジョンなど）を扱う点にある。また、対象となる項目は、組織的・社会的・心理学的状況を含む、グローバルな環境変化という大きな視点から描かれる。

ヒューマンインターフェース

HCIの分野では、80年代半ばからシナリオが注目を浴びはじめ、SIGCHI Bulletinが1992年に特集号を出した。この分野では、主に設計と評価においてシナリオがよく用いられる。

著者の1人は、1994年から、教育用グループウェアシステムを設計・開発するプロジェクト、LiNC（Learning in Networked Communities）を推進している³⁾。LiNCプロジェクトでは、3年以上の長期にわたって、システムのユーザである現場の中学生・高校教師とともにシステムの設計を進めている。ここでは以下のような観点から、システムの仕様を書く代わりに物語体の文章のシナリオを記述している。

- (1) 技術者でない中学・高校教師が記述・理解しやすい。
- (2) 設計会議におけるブレインストーミングの材料としてシナリオを用いる。

(3) 構築前の新しい技術や機能を、現実のもののように想像する作業を行う。

(4) 実装前にある程度のシステム評価を行うために、簡易プロトタイプとしてシナリオを用いる。

LiNCプロジェクトでは、中学生がネットワーク上に作られた仮想学校で共同学習をする様子を描いた基本シナリオから、多くのシナリオを導出・作成する作業を通して、システムの設計・開発を行っている。

HCIの分野では、設計以外に評価においてシナリオがよく用いられる。システムのユーザビリティ（使いやすさ）を評価するために、ユーザとしての被験者を対象に系統的な試験（ユーザビリティテスト）を行う。このとき、被験者が行う作業を設定するために、シナリオが用いられる。

ユーザビリティテストでのシナリオの例を挙げる。

(1) 所属会社の標準電子メールソフトウェアとしてEudoraを使うことになった。ソフトウェアの初期設定をし、kgo@acm.org宛にテストメールを送信せよ。

(2) 上司から会議用の書類を至急印刷するように頼まれた。Word 98で現在開いているファイルを4部印刷せよ。

ユーザビリティテスト用のシナリオは、ふつう物語体の文章形式で与えられる。この場合のシナリオには、簡潔性と具体性、そして非曖昧性が求められる。

要求工学

要求工学とは、システムの要求獲得・分析作業の工学化を目指した研究分野である。したがって、作業手順や成果物を系統的・定量的に扱う。要求工学での主な作業は、ユーザや顧客の要求を分析し、それを満足するようにシステム外部動作仕様を定義することである。そのため、システムの視点に重きを置いたシナリ

オを利用することが多い。この場合、システムの機能を明確にするため、コマンドやキーストロークレベルの詳細なイベントをシナリオに書く。

シナリオを用いた要求工学で、現在最も重要な役割を果たしているのは、欧州のCREWS（Cooperative Requirements Engineering With Scenarios）プロジェクトである⁵⁾。CREWSプロジェクトでは、シナリオを用いた支援ツールと手法の構築を目標にしている。具体的には、マルチメディア記録システムを用いた要求分析作業支援、自然言語理解による要求獲得支援、要求のアニメーションと比較によるシナリオ検証支援を対象にしている。初期のCREWSプロジェクトでは、シナリオを分類するためのフレームワークを定義した。ここでは、形式、内容、目的、ライフサイクルという4つの視点からシナリオを分類し、これを基準にシナリオに基づく各種の手法の比較を試みている。

オブジェクト指向分析・設計

オブジェクト指向分析・設計では、シナリオをオブジェクトの抽出に利用する。そして、システムの抽象モデルの構築を目標にしている。

近年この分野で中心的話題となっているUML（Unified Modeling Language）では、ユースケース（Use Case）とシナリオという概念を、その中に導入している^{☆2}。ユースケースによるモデル化では、外部との相互作用によってシステムを記述するブラックボックスの視点がとられ、アクターとシステムとの相互作用を順番に記述することでモデル化を行う。ただし、ここでのアク

☆2 正確には、ユースケースとシナリオとは同一概念ではない。前者は取り得るすべてのイベント列を表現するが、後者ではその一部を表現しているだけである。また、前者は形式的な操作を指向しているのに対し、後者は非形式的な操作を指向している。すなわち、ユースケースアプローチにおいてシナリオに対応するのは、ユースケース・インスタンスである²⁾。

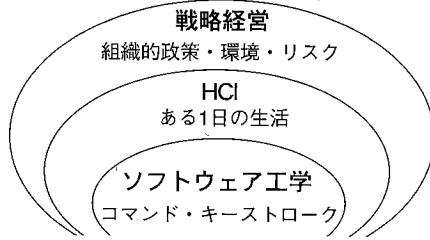


図3 シナリオを用いる分野間の関係。ソフトウェア工学は、要求工学とオブジェクト指向分析・設計を含む。

ターは必ずしもユーザである必要はない、対象システムと相互作用する他のシステムの場合もあり得る。

4分野の関係

上記の4分野におけるシナリオの役割の違いは、シナリオが対象とする範囲に明確に表れている。戦略経営では、年月を基準にしてシナリオを作成し、そこでは、組織的政策・環境・リスクを扱う。逆に、ソフトウェア工学（オブジェクト指向モデリングと要求工学）の分野では、コマンドやキーストロークレベルの機能に関するシナリオを作成する。HCIのシナリオは、これらの間に位置し、対象とするシステムに関して、ある1日の生活（Day-in-the-life）のシナリオを記述する（図3参照）。

以上の関係を考慮し、より高品質のシステムを設計・開発するには、できるだけ多様なシナリオを作成することが有効だと思われる。すなわち、システムが何をどうするかというソフトウェア工学（要求工学、オブジェクト指向分析・設計）での視点と、ユーザがそれで何をし、どう使うかというHCIでの視点、そして、それが社会的・組織的にどう受け入れられ、どう影響を与える（与えられる）かという戦略経営での視点で、シナリオを作成し分析することにより、包括的なシステムの品質向上が期待される。

☆3 これらの課題に対するアプローチをWebページ<<http://metatron.esi.yamanashi.ac.jp/go/scenario.html>>にまとめたので参照してほしい。

シナリオに基づく設計・開発の工学化をめざして

シナリオはユーザを中心とした記述である。したがって、シナリオを書くことで、ユーザの視点をシステム設計に導入することができる。その結果、システムの使いやすさの観点から、構築されるシステムの品質向上が期待される。しかも、シナリオを開発工程に導入するのに特別なツールは不要ない。たとえば物語体の文章でシナリオを記述するのであれば、普段から用いられている文書作成ツールがあれば十分である。すなわち、比較的（経済的・心理的・文化的に）低い導入コストで、シナリオに基づく設計法が使えるかどうか、実際に開発プロジェクトに導入して試験・評価することができる。

シナリオに基づく設計法の工学化をめざした研究は始まったばかりである。特に、これまで分野ごとに個別に行われていた研究を統合し、ソフトウェア設計・開発に有効な方法論として確立するためには、解決しなければならない課題が数多くあ

る。工学的に重要な課題を列挙すると以下のようになる☆3。

- どうすればシナリオをうまく管理することができるのか？
- どうすればシナリオ作成作業を軽減することができるのか？
- どうすればシナリオの質を統一できるのか？
- どうすればシナリオと仕様とをうまく組み合わせることができるのか？
- どうすればシナリオをうまく再利用できるのか？
- どうすれば設計・開発の現場に積極的に導入できるのか？

さらに、シナリオを設計活動から得られた成果物とみなし、その分析を行えば、人の創造的な設計活動の性質を明らかにできるかもしれない。この意味で、シナリオに基づく設計法の研究は、「何かを創り出す」という人間の持つ本質的な特徴を解明する可能性を秘めている。

参考文献

- 1) 黒須正明, 伊東昌子, 時津倫子: ユーザ工学入門, 共立出版, 東京(1999).
- 2) Carroll, J.M. (ed.): Scenario-Based Design: Envisioning Work and Technology in System Development, John Wiley & Sons, New York, NY (1995).
- 3) Carroll, J.M., Rosson, M.B., Chin, G. and Koenemann, J.: Requirements Development in Scenario-Based Design, IEEE Transactions on Software Engineering, Vol.24, No.12, pp.1156-1170 (1998).
- 4) McGraw, K. and Harbison, K.: User-Centered Requirements: The Scenario-Based Engineering Process, Lawrence Erlbaum Associates, Mahwah, NJ (1997).
- 5) Rolland, C., Achour, C.B., Cauvet, C., Ralyte, J., Sutcliffe, A., Maidan, N.A.M., Jarke, M., Häumer, P., Pohl, K., Dubois, E. and Heymans, P.: A Proposal for a Scenario Classification Framework, Requirements Engineering, Vol.3, No.1, pp.23-47 (1996).
- 6) Wack, P.: Scenarios: Uncharted Waters Ahead, Harvard Business Review, Vol.63, No.5, pp.72-89 (1985).
- 7) Weidenhaupt, K., Pohl, K., Jarke, M. and Häumer, P.: Scenarios in System Development: Current Practice, IEEE Software, Vol.15, No.2, pp.34-45 (1998).

(平成11年11月30日受付)