

時間的变化を伴う複雑な事象の表現手法に関する研究

高嶋 章雄[†], 蔵川 圭[†], 山本 恭裕^{†§}, 中小路 久美代^{†§}
e-mail: { akio-ta, kurakawa, yasuhi-y, kumiyo }@is.aist-nara.ac.jp

[†]奈良先端科学技術大学院大学 情報科学研究科

[‡]株式会社 SRA ソフトウェア工学研究所

[§]科学技術振興事業団 TOREST

概要: 情報を視覚化することにより人間はそれをより直感的に理解することができるため, グラフィックスを用いた様々な表現手法が提案されてきた. 本稿では時間的变化を伴う事象をアニメーションを用いて表現し, 情報を得る上で重要となるインタラクションについて論じる. ケーススタディとしてオブジェクト指向プログラミングによるクラスライブラリの進化を表すアニメーションを制作した. ユーザ観察を通して, (1)時間軸のコントロール, (2)差分の表現, (3)臨場感・没入感, (4)情報の絞りこみ, (5)時間軸のメタビュー, という 5 つの側面が, アニメーションを用いたインタラクションにおいて重要であることがわかった.

Representing and Interacting with Complex Data with Temporal Variations

Akio TAKASHIMA[†], Kei KURAKAWA[†], Yasuhiro YAMAMOTO^{†§}, Kumiyo NAKAKOJI^{†§}

[†]Graduate School of Information Science, Nara Institute of Science and Technology

[‡]Software Engineering Laboratory, Software Research Associates, Inc.

[§]TOREST, Japan Science and Technology Corporation

Abstract: Graphic representations, such as charts or diagrams, aid humans to grasp the meaning of information more easily. Computers have been used to visualize information, and a number of graphical representation methods have been explored. This paper describes a representation of complex data with temporal variations by using animation. By observing an expert interacting with the representation, we have identified five concepts that are important for using animation to visualize such complex data: (1) a control of the time scale, (2) differential representation, (3) a feeling of immersion, (4) filtering out information, and (5) a meta view for the time scale.

1 はじめに

古くから, 様々な情報をより直感的に理解させることを目的として, 絵やグラフなどの視覚的な表現が多く用いられてきた[1]. 近年では, 計算機の普及およびその性能の向上に伴い, 計算機上で多種多様な視覚的な情報表現が可能となってきた. しかし一方で, それぞれの表現方法がもつ効果をよく理解せずに利用することによって, 情報が歪められて伝わったり, 焦点が曖昧になってしまうという問題があ

る[2]. より高度な表現手法を上手く利用するためのノウハウを知ることが, 情報視覚化における初期段階での基本的かつ重要な要素であると言えよう.

情報視覚化に関しては, 例えば地図のように実在の空間(地形)を 2 次元あるいは 3 次元の仮想的な空間にマッピングして表現することは極めて自然であり, 空間を拡大・縮小して表現することにより実世界では捉えにくい情報を容易に得ることができる. 同様に実際の時間も仮想的な時間にマッピングして表

現することで、時間を縮めることや止めることさえ可能となる。本研究は時間的な変化を伴う事象を、異なる時間軸へとマッピングすることによる表現手法を探ることを目的とするものである。

Disney に代表されるアニメーション映画 (animated cartoon) の制作に関しては、効果的な動きを表現する知見が報告されているが[3]、実世界に存在しない情報を視覚化して表現する際のアニメーションの利用に関するノウハウの報告は少ない[4]。

本稿では、まず、時間軸をもつ情報を視覚化するためのアニメーションによる表現に着目し、既存のアニメーションの特徴と利用のされ方について説明する。次に、オブジェクト指向プログラミングによるソフトウェア開発におけるクラスライブラリの進化を対象としたケーススタディについて報告し、考察する。

2 アニメーションの利用

コンピュータ上でアニメーションを用いる既存の研究の多くは以下の 2 つの側面から大別することができる。

- (1) 対象の変化をアニメーションで表現するもの
- (2) 視点の変化をアニメーションで追従するもの

(1)の分類で代表的なものとして、アニメーション映画のように人間やネズミ、ボールといった実世界に存在するものの動きを CG でリアルに、時には誇張して表現するものがあり、効果的かつアニメーションでしかできない表現に関するノウハウも提案されている[5]。また、アニメーション映画に限らず、VRなどの分野でも対象の特徴的な動きを表すテクニックとしてアニメーションが多く用いられている。

次に(2)の分類は、ディスプレイ上に表現されている対象をインタラクティブに操作する際、あるいは対象を見る視点を変化させる際にアニメーションを用いるものである。情報視覚化の分野で、情報全体の構造をおおまかに表示しながら注目点付近を詳細に表示するための focus + context の技術に関する研究が多く行われている[6]。そのほとんどは、視点を移

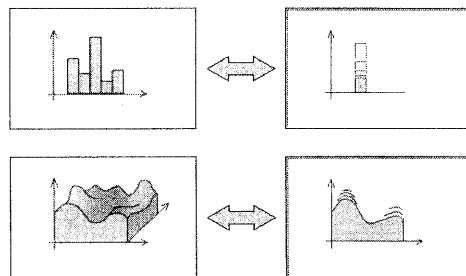


図 1: 静的表現と動的表現の対比

動する際に、前の視点から次の視点へのスムーズな変化をアニメーションで表現することによって、自分が注目している点と全体との関係を見失わないようにすることを実現している。

(1)の分類の代表的なものとして、実際に存在する物体の動きをアニメーション化することを挙げたが、棒グラフや折れ線グラフなどのように、もともと目に見えない情報を可視化したものの変化をアニメーションで表現することも可能である[7]。しかし、このようなアニメーションの利用に関してはノウハウがほとんどないのが現状である。図 1 では、2次元のグラフと1次元+アニメーションのグラフの対比、および3次元のグラフと2次元+アニメーションのグラフの対比を表している。ほぼ同じだけの情報を扱っていたとしても、見るものにとって非常に異なる印象を与えるであろう。グラフに限らず、視覚化された情報をアニメーションを用いて表現する場合には、その長所や短所を十分に理解した上で利用することが重要である。

情報視覚化の分野においては、プログラムの実行状態や、アルゴリズムなど時間的な変化を伴う情報を表現する研究も行われている[8][9]。本研究ではこのような時間軸に沿った情報の変化を、仮想的な時間、すなわちアニメーションで表すことに着目した。

(2)の分類で挙げた focus + context の技術は、情報の構造自体を対象とするもので、言い換えれば空間的な focus + context であると言える。これは一般に構造の決まった静的な情報に対して用いられる技術

であるが、情報の時間的な変化をアニメーションで表現した場合も同様に、「時間的な focus + context」が存在し得ると考えられる。アニメーションで表現された流れの意味を理解する際、ただ単純に制作されたアニメーションを再生するだけではなく、自由に時間を区切って再生したり、再生中に少し前の状態やこれから起こる変化などを提示することが、全体を通しての情報の理解に繋がるであろう。時間的な focus + context を表現することができれば、情報をより正確に、詳細まで理解することが可能であると考えられる。

以上のことを考慮し、時間的な変化を伴う情報表現の可能性を探るための第1段階として、3次元棒グラフによる単純なアニメーション作成環境を構築した。続く3章で、作成したシステムおよび実際に利用したデータについての説明を行い、4章でユーザが制作されたグラフアニメーションとどのようにインタラクションを行うかの観察結果について述べる。

3 ケーススタディの概要

3.1 システム

本システムで構築するアニメーションは3次元棒グラフを動的に表現するもので、アニメーションの1コマ1コマがある時刻における情報を示している(図2左)。さらにその前後の時刻の値もオーディオ製品に見られるピークメータに似た表現で提示することができ、わずかではあるが時間的な context を保持できる。ウィンドウ内に表示されたグラフは、再生しながらでもマウス操作によるインタラクティブな視点

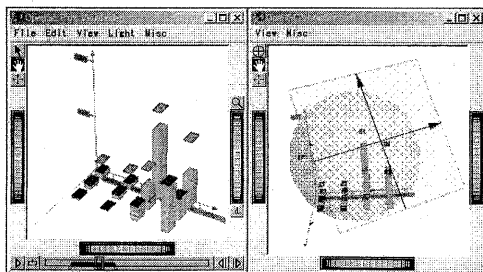


図2: アニメーションの例

(対象)の変化が可能で、ズームやパン、投影法の変更も行うことができる。表示されているウィンドウの持つ視点の位置や視野角、視線の方向などを捉えたメタなビュー(ビューポート)を別ウィンドウで開くことができ、対象の見方に対する context を保持することも可能である(図2右)。また、時間の流れはスライドによってコントロールでき、再生スピードの変更や範囲を指定したループ再生などの機能を持つ。

なお、本システムは、オープンソースソフトウェア3次元グラフィックライブラリ「じゅん(Jun)」(<http://osb.sra.co.jp/Jun/>)を利用して構築した。

3.2 利用データ

時間軸を伴う事象の題材としてオブジェクト指向ソフトウェア開発を例に取り、そのクラスライブラリの進化を、システムを利用したアニメーションで表現した。ソフトウェア開発ではバージョンが不規則な時間間

Version	date	Class	Instance Methods	Class Methods	
Jun005	1996/10/29	19,31,38,000	105	2401	810
Jun006	1996/10/30	19,36,27,000	107	2445	836
Jun007	1996/10/31	17,29,00,000	107	2456	853
Jun008	1996/10/31	20,29,37,000	107	2458	853
Jun009	1996/11/17	21,33,31,000	110	2636	802
Jun010	1996/11/24	23,29,25,000	110	2633	806
Jun011	1996/11/28	11,41,39,000	114	2696	825
Jun012	1996/11/28	19,13,49,000	113	2687	810
Jun013	1996/11/29	17,00,41,000	113	2687	819
Jun016	1996/12/23	18,55,57,000	118	2876	1015
Jun019	1996/12/27	1,47,58,000	118	2880	1015
...
Jun242	2000/3/15	19,02,27,000	696	14070	5233
Jun243	2000/3/16	23,23,08,000	696	14071	5235
Jun244	2000/3/17	18,22,11,000	696	14071	5236
Jun245	2000/5/6	20,31,23,000	640	14186	5209
Jun246	2000/5/8	10,34,07,000	640	14186	5291
Jun247	2000/5/18	19,37,38,000	640	14212	5293
Jun248	2000/6/28	22,02,20,000	640	14212	5293
Jun249	2000/6/28	15,10,22,000	640	14215	5293
Jun250	2000/6/30	14,23,08,000	640	14215	5293
Jun251	2000/7/1	9,34,45,000	640	14215	5293
Jun252	2000/7/20	11,05,51,000	640	14217	5293

図3: 表による表現

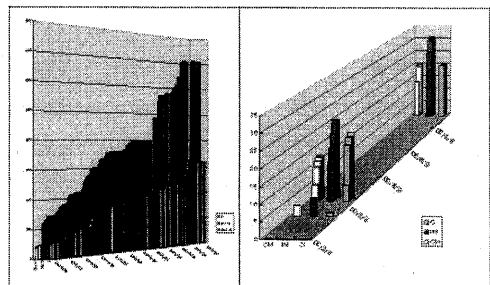


図4: 空間の奥行きを時間軸にしたグラフ

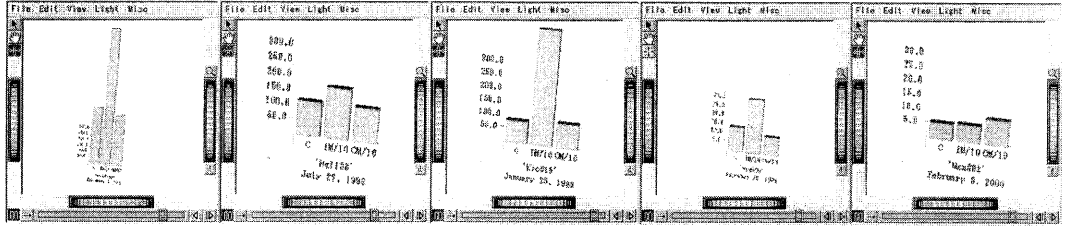


図 5: プロジェクト毎のグラフアニメーション

隔でリリースされ、かつそれぞれのバージョンが単体で多くの複雑な情報を有している。これらのバージョン間の関係を理解することで、開発の歴史、あるいは進化の動向を把握することが可能となるため実験の題材として適当であると思われる。

今回対象としたのは、Smalltalk 用クラスライブラリの 5 つの開発プロジェクト(内 1 つはベースとなるクラスライブラリ)で、いくつかのプロジェクトは並行して開発が進められた。また、この内 1 つのプロジェクトとベースとなるクラスライブラリの進化は現在もお続けている[10]。

利用したデータはそれぞれのプロジェクトのクラスライブラリにおけるクラスの総数、インスタンスメソッドの総数、クラスメソッドの総数、およびバージョンナンバーとリリースされた日時である。これらの情報を、表形式で表したもの(図 3)、および空間の 1 軸を時間軸とした 3 次元棒グラフで表したもの(図 4)を示す。ベースとなるクラスライブラリ開発プロジェクトは、4 年以上の歳月をかけて 300 以上のバージョンがリリースされており、どちらの表現手法でも、全バージョンを通しての開発の流れや、個々のバージョン内でのデータの間関係を同時に読み取ることは困難である。

制作したアニメーションにおいては、1 コマが実時間での 1 日を表し、その時点でのバージョンにおけるクラス、インスタンスメソッド、クラスメソッドの総数が棒グラフで示され、同時に前バージョンおよび次バージョンでのそれぞれの総数も表示される。また、現在表示されているクラスライブラリのバージョンナンバーとリリースされた日付・時刻が、グラフ下部に表示される。さらに、制作した 5 つの開発プロジェク

トのグラフアニメーションを、日付を基準に同期して再生可能とした(図 5)。

4 ケーススタディにおけるユーザ観察 およびインタビューに基づく考察

制作したグラフアニメーションを、実際にクラスライブラリ開発に携わったメンバーのひとりに見せ、その場でインタビューおよびディスカッションを行った。被験者には、クラスライブラリの進化を、アニメーションとして表現したのを見て気づいたことや感じたことを発話してもらい、その発話プロトコルと操作履歴を記録した。以下に示す発話プロトコルでは、文頭に # ががついているものがインタビューアの発話、何もついていないものが被験者の発話である。

被験者の操作の観察結果およびインタビューから、以下の 5 つの側面に着目したことがわかった。

- (1) 時間軸のコントロール
- (2) 差分の表現
- (3) 臨場感、没入感
- (4) 情報の絞りこみ
- (5) 時間軸のメタビュー

またドメインに特化した興味深い結果も得られた。

4.1 インタラクションにおける 5 つの側面

アニメーションを用いたインタラクションにおいて重要となる 5 つの側面それぞれについて、得られた発話やディスカッションをもとに考察を行う。

(1) 時間軸のコントロール

被験者はアニメーションの再生を制御するスライダを多用し、自由に時間をコントロールしていた。特にある一定の時間幅(スライダ幅)で、時間を行ったり来たり(早送り・巻き戻し)することで、値の差の変動

を繰り返し見ようとする傾向があった。次に示すインタビューからも、時間軸をコントロールすることが重要であることがわかる。

ゆっくりした再生だからいいのでしょうか？普通に再生されてるときよりはちょっとづつ動かされてるときの方が…

そうそう。これが通常に再生されているときよりも、

自分でコントロールできるから。

自分でコントロールできると、これ(バージョン情報)とこれ(各要素の値)との関連を見つけやすいですよ。

だから、自分で時間のコントロールができるのがいいんですね。

うん、たぶん。このバージョンを見るときはそれが一番いいような気がするなあ。

絶対、でも、コントロールしたいんですね。

コントロールしたいよねえ。なんか、アニメーションの良さって、これ、流すよりも、自分でコントロールできるところに良さがあるのかもしれない。

ユーザ観察全体を通してこのような時間のコントロールに関する発話が頻繁に行われ、時間を自由に操作したいという要求が非常に強いことがわかった。時間的な変化をアニメーションで表すことの最大の利点は、データが持つ時間の流れの速さにとらわれず任意の速度で再生ができることであるが、さらに一時停止させたり、データに大きな変動が見られる付近の時間を進めたり遅らせたりしつつ繰り返し見ることによって、より詳細まで情報を理解することが可能となる。時間をコントロールして見ることによって、全体の流れや、ある時刻の周辺でどんな動きをするかを見つけることができ、ただ単に一定のスピードで再生するよりも、データ間の関連を見つけやすいことがわかる。

(2) 差分の表現

表示されたデータの値と前バージョンおよび次バージョンの値を見ることにより、その間の差分が大きい場所を繰り返し見つけようとしていた(図6)。

赤とか青の影(前後のバージョンの値)がビヨビヨでてる甲斐があるんですか？

あるある。だって前のと必ず比べられるもんね。

さっき減ったか伸びたかを、ですね。

ほっ、ここかあ。いつやねんて。

ああ、一気に差が増えるところですか。

やっぱりこの差を見つけにきますよ。

これはexcel(による奥行きを時間軸とした3次元グラフ表現)のまほうがわかりやすいかも知れません。

でも、excelってこんな風にできないじゃないですか。こんな差を見るような感じは。

自分で見よう見ようとしてシミュレーションしないと。

そうそう。後ろのやつと、この前のやつとのグラフの長さを比べて、自分で見なければいけないけど。でもこれは、なんというか1ドラッグで、これだけでとりあえず見えるから。

時間を自分でコントロールするだけでなく、前後の値も表示されていることが、大きな変化が現れる時刻を容易に特定するのに役立っていた。アニメーションを用いずに空間内の軸のひとつを時間軸とする場合は、注目しているデータと前後のデータを意識的に見比べてシミュレートし、差分をイメージする必要があるため、認知的負荷が大きいとも言える。1スナップショット内にその時点での値に加えて、前後の時刻の値も付加することによって、アニメーションの流れを捉えたり、値の増減を予測するのに役立つと考えられる。

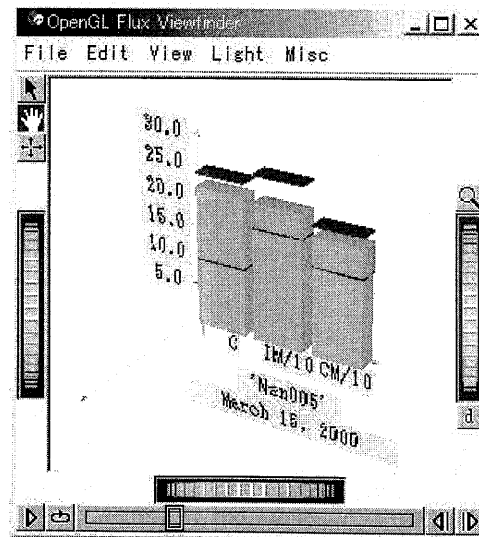


図6: 差分の表現

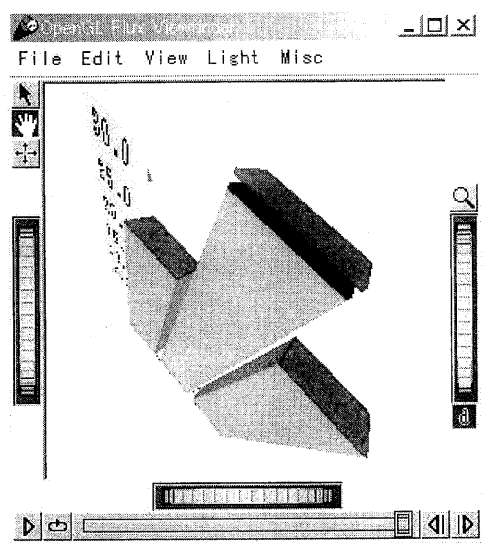


図 7: グラフ上方からの透視投影

(3) 臨場感・没入感

透視投影による表現で棒グラフを上から見下ろすような見方をすると、数値の正確な値はわからなくなるものの、値の変動や、他の要素との関連をより強く感じられた。かぶりついたような視点になり、手前に来るものほど大きく見え、変化が顕著に現れた(図 7)。

*Perspective*をかけて上から見てのほうが認識しやすい可能性がある。どうしてもチャートやグラフになると、横から見ってしまうっていうのを学校教育で習いすぎてますよね。で、伸びてくるのはパードビューで見て、グイーンって自分に近づいてくるのが *perspective* であると、近いとこが大きくなったとかちいさくなったとか... そういうほうが良さそう...

手前にあればあるほど大きくなるから...なんだろう、顕著に見える...

#VRML のグラフの中を歩く、没入感みたいなのに似てますね。

そうだね。

#いま3Dのビューのコントローラは完全に自分はこのに止まってて、相手を動かす感じですよ

アニメーション停止時、再生中に拘らず、空間内の視点を移動したり投影法を変更することにより、臨場感あふれる表現を“体感”することが可能であった。これにより、具体的な値はわかりづらくなるものの、他の要素との関係を感覚的に捉えるという点では重要な要素であると言える。

(4) 情報の絞りこみ

excel (による奥行きを時間軸とした 3 次元グラフ表現) でなにいけなかったっていうと、いらん情報まで見えすぎちゃうんですね。自分がフォーカスしているところは、ほんの狭い範囲でフォーカスしてるのに *excel* だと、全部が見えちゃってほかくに注意を奪われるけど...

#全時間が一度に見えてるのがまずいんですか

たぶん。その全体像をつかみたいときは *excel* のグラフでいいのかも知れないけど

この発話からもわかるように、時間軸を空間上の 1 軸にすると全体像をある程度把握することができる。しかしその反面、狭い範囲をフォーカスして見たいときに、多くの情報が見えすぎて情報を理解しづらいと考えられる。

図 5 に示した通り、5 つのプロジェクトの変化をそれぞれのウィンドウで表示したが、同期再生する際にも、

ひとつの(ウィンドウ)の中にあるよりも、別々のほうが良いかも... この枠っていうかスコープが非常に大事で、全部をひとつのスコープの中に入れてしまうと、余計なものが見えすぎる。

という意見が得られた。一度にどの程度の量の情報を見せるかを考慮することが、効果的に情報を理解する上での不可欠な側面である。

(5) 時間軸のメタビュー

システムには視点や視野角、視線の方向を捉えるための空間的なメタビューを表示する機能がある(図 2 右)。

これ(ビュー)が自分でフォーカスしてると思うと、これ(ビューポート)がもうひとつメタへ上がったコンテキストを押さえているじゃないですか。やっぱアニメーションにもそういう視点あるんじゃないですかね。なんかそんな気がします。そうするとよくわかりますよね。

このように、アニメーションにも時間的な context を保持するためのメタビューが必要ではという発話を得られた。日付やバージョンは一種のメタビューと見ることができが、さらに現在動かしている範囲がどこであるかを示すスライダ上に日付やバージョンを示すことで、全体のどの部分を再生しているのかを容易に捉えることができるであろう。しかし、この

context は時間軸に対する context ではあるが、時間的に変化するデータに対する context ではない。今回の実験ではデータの値を表示したときに同時にその前後の時刻での値も表示したことから、わずかではあるがデータ自体の時間的な context を保持できたと言えるものの、アニメーション全体の流れをより大局的にとらえるメタな表現が必要であると考えられる。

4.2 ドメインに特化した観察結果・考察

観察結果として得られたものの内、ソフトウェアの開発に特化した知見がいくつか発話として得られた。まず第1に、リリースするバージョンのナンバーが区切りのいい数字の場合は、無理にたくさんの機能を組み込んでしまい、その後不備が見つかってやり直しをする(クラス数やメソッド数が減少する)という変化が見て取れた。

また、十分に設計して開発を進めた場合、最初からどんなクラスが必要であるかキャストイングが決まっているため、まずクラスが増えてからメソッドが増加する。この変化をアニメーションを通して見ることができ、いくつかのプロジェクトが被験者自身の作り方であることが確認された。被験者が直接開発に関係していないプロジェクトの進化の場合は、明らかに被験者の作り方ではないと断言した。

これは、インプリが迷い箸ってやつですよ。

#なんでわかるんですか？

つまり、クラスとメソッドの増え方が同期してるんですよ。この作り方っていうのは基本的にちゃんと設計してない。

#設計していたら？

最初クラスだけがが増えて、そのあとメソッドが激増してくる。これって、オブジェクト指向のXP(extreme programming)になる作り方なんですけど。

これだと、trial&error に近い、えーい、やってみるかって、やってる。

#それはアニメーションを見てわかるものですか？

わかりますね。だって同期してるもん。クラスの数とメソッドの数が。

#片方が増えるときは、もう一方も増える。

うん。だってNsn(被験者自身が開発に携わったプロジェクト名)なんか一番最初にクラスが増える、うん。そのあと、メソッドが増えてくる。だよな。そうだね。明らかに。つまり、設計してることですよ。

あ、でも、こういうのって大事なんじゃないかなあ。こういうのって伝えるに良かったんですよ。むしろソフトウェア工学的にオブジェクト指向の方法論教えてても。

この発話から、アニメーションを用いて表現したことが、クラスやメソッドのような要素間の関連を時間軸に沿って把握するために役立ったと言える。

さらに、すべてのプロジェクトのアニメーションを同期させて再生した際、並行して進められたプロジェクトが影響しあっている様子がわかった。例えばベースとなるクラスライブラリが初期の段階では、別のプロジェクトで必要となった機能を実現するために、ベースのライブラリが頻繁に更新され、双方の変化が連動していることを確認できたが、比較的新しいプロジェクトでは既に基盤が整っているため、そのプロジェクトの影響はベースのライブラリに及びにくいことが見て取れた。

4.3 全体を通しての操作傾向

実験全体を通して、被験者は、

- (1) アニメーションをコントロールするスライダを自由に操作し、クラスやメソッドの数が急激に増減する場所を探し出す
- (2) その後同時に表示されているバージョン情報や日付を見て、そのころに何を開発していたか、どんな状況であったかを思い出す
- (3) さらに、思い出すと同時にその先の変化も予測し、アニメーションを進ませることで確認をする

という一連の動作の繰り返しを行うことが観察された。アニメーションを用いた表現が、データの変化を捉えるきっかけを作る役割を果たしたと考えられる。

5 今後の課題

本稿では時間的な変化を伴う事象をアニメーションを用いて表現し、そこから情報を読み取る際のインタラ

クシオンにおいて、時間軸のコントロール、差分の表現、臨場感・没入感、情報の絞りこみ、時間軸のメタビュー、という5つの側面が重要であることを述べた。これらの側面を、汎用的なアニメーション表現の枠組みにどのように取り入れていくかは今後の課題である。

特に、被験者の意見としても得られたように、アニメーションを用いて時間を表現する際には、時間的な context を表現することが空間的な context 同様に必要とされていることがわかった。context は空間内での見方や、全時間に対する現在の時間位置を示すものだけではなく、データ自体、あるいはその変遷を示す context も重要であることがわかった。空間的な focus + context と時間的な focus + context をひとつのウィンドウ内に表現することができれば、アニメーションの1コマが持つ意味と全体の流れが持つ意味(図8)を理解する上で、重要な役割を果たすと考えられる。

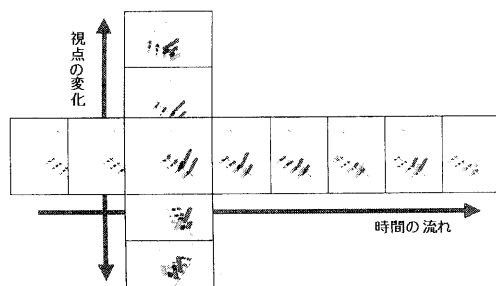


図8: 1コマが持つ意味・時間の流れが持つ意味

謝辞

本研究における理論的枠組みの構築、システムの設計、評価、および考察を行うにあたってご協力を頂いた青木淳氏、大平雅雄氏、および B. Reeves 氏に心より感謝の意を表す。なお本研究の一部は、新エネルギー・産業技術総合開発機構・新規産業創造型提案公募事業の援助によるものである。

参考文献

- [1] Edward R. Tufte, Visual Explanations, Graphics Press, Cheshire, CT, 1997.
- [2] Donald A. Norman, Things That Make Us Smart, Addison-Wesley Publishing Company, Reading, MA, 1993.
- [3] The Animator's Workbook, Watson-Guptill Publishing, Toronto, Canada, 1988.
- [4] Ivan Herman, Guy Melançon, and M. Scott Marshall, "Graph Visualisation and Navigation in Information Visualisation: a Survey," IEEE Transactions on Visualization and Computer Graphics, Vol.6, No.1, pp.24-43, 2000.
- [5] John Lasseter, "Principles of Traditional Animation Applied to 3D Computer Animation," SIGGRAPH '87, Computer Graphics, Volume21, No.4, pp.35-44, 1987.
- [6] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman, Readings in Information Visualization: Using Vision to Think, Morgan-Kaufmann Publishers, San Mateo, CA, 1999.
- [7] William Wright, "Information Animation Applications in the Capital Markets," Proceedings of InfoVis '95, IEEE symposium on Information Visualization, pp.19-25, color plates 136-137.
- [8] Hideki Koike, Tetsuji Takada, "Visualinda: A Framework and A System for Visualizing Parallel Linda Programs," University of Electro-Communications, IS Technical Report, UEC-IS-1997-8.
- [9] Marc H. Brown and Marc A. Najork, "Algorithm animation using 3D interactive graphics," Proceedings of the sixth annual ACM symposium on User interface software and technology, 1993, pp.93-100.
- [10] Atsushi Aoki, et al. "A Case Study on the Evolution of Jun: an Object-Oriented Open-Source 3D Multimedia Library," Proceedings of International Conference on Software Engineering (ICSE'2001), Toronto, Canada, 2001 (to appear).