
Time-Machine Computing: 時間指向ユーザインタフェースの提案

Time-Machine Computing: Towards Time-centric User Interfaces

暦本 純一*

Summary. This paper describes a time-centric approach for organizing information on computers, called “Time-Machine Computing.” A system based on this approach allows a user to visit the past and the future states of computers. When a user needs to refer to a document that he/she was working on at some other time, he/she can travel in the time dimension and the system restores the computer state at that time. The combination of spatial information management of the desktop metaphor and time traveling allows a user to organize and archive information without being bothered by folder hierarchies or the file classification problems. Time-Machine Computing also proposes a mechanism for linking multiple applications by exchanging time information. This paper describes the key features of our proposed approach, as well as a time-machine desktop environment called “TimeScape.”

1 はじめに

文書ファイルなどの電子情報を整理・管理するためのインタフェースはコンピュータの基本機能のひとつであり、コンピュータシステム全体の使い勝手を大きく左右する。従来は、フォルダ階層やファイル名による分類を主体とした方式が使われてきた。しかし、コンピュータの利用層が技術者・専門家から家庭などでの利用者に拡大するに従って、より直観的で親しみやすい方式が求められるようになってきている。こういった利用者が主に扱う電子情報は、写真・手紙・メモ・WWWの素材などであり、厳密に分類して階層的に管理する方式は向いていない。

Maloneらは、人々の(現実の)机上での作業を観察し、空間的な位置による文書管理が多く利用されることを見出した。この知見にもとづき、Roomsと呼ぶ空間的な情報管理を主としたシステムを提案している[10]。また、Manderらはデスクトップ上でアイコンを積み重ねて管理する‘pile’メタファー[11]を提案している。情報を厳密に分類せずに、空間的に配置する方法は、現状のデスクトップ環境でも日常的によく見かける光景である。たとえば、作業中のファイルを(一時的にであれ)直接デスクトップに置く利用者は少なくないだろう。これらのアイコン群は、単に情報の格納という意味だけではなく、作業コンテキストを表現する手段としても有効である(たとえば、次の日に手掛けるべき文書をデスクトップに出しておく、など)。この効果を強化する目的で、ファイルアイコン以外にも、電子的な付箋紙(ポストイット

* Jun Rekimoto, ソニーコンピュータサイエンス研究所インタラクションラボラトリー

ト)が使われることがある。また、アイコンを空間的に配置して、ゆるやかなグループを構成することも可能である [4]。

これらの「空間配置指向」の方法と対照的に、Freemanらはファイルを時間順に整列・管理するシステム Lifestreams[3]を提案している。これは「超整理法」[17]に似た発想で、ファイルの内容やタイトルなどに関わらず、すべてを時間順の1次元の列に並べて管理する。「空間配置指向」の方法はスケーラビリティに問題があるので時間順方式が勝っていると主張している [2]。

本研究では、この両者を排他的ではなく相補的なものと捉え、個々のファイルを時間順に配列するのではなく、「コンピュータ環境(狭義にはデスクトップ環境)」の状態そのものを時間順に管理する方式を提案する。この方式は、いわば「コンピュータ環境のタイムマシン」を構築することに相当する。もしコンピュータの状態を任意の時刻に戻せるとするならば、とりあえず必要なくなったファイルは画面から消してしまっても構わなくなる。あとで再度必要になったときは、「そのファイルについて作業していた日」にまで遡ってファイルを取って来ればよい。デスクトップに直接アイコンを置く方式でも、スケーラビリティの問題を回避できる。この発想は、GUIのみならず一般のコンピューティングにも適用できる。たとえば「任意時点でのファイルシステムの状態」が再現できるようになれば、特別なバージョン管理コマンドで管理しなくても、任意の時点のファイル内容を再現・比較することができる。このような、時間移動の概念を中心として情報管理を行なう方針を **Time-Machine Computing** と呼ぶことにする。具体的には、以下のような特徴をもつユーザインタフェースである。

作業履歴の蓄積: 利用者がコンピュータに施した変更を記録して、コンピュータの状態を再現可能にするのが Time-Machine Computing の基本的な発想である。「コンピュータの状態」には実現レベルで様々な段階がある。画面データを(ビットマップとして)保存する方法や、コンピュータのメモリをまるごと保存することも考えられるが、より現実的には、ファイルシステムの状態のみを保存する、デスクトップ上でのファイルやアイコンの履歴を保存する、など、部分的な記録に留めることも考えられる。

時間移動による作業環境の再現: 蓄積された情報を使って、過去の任意の時点でのコンピュータ環境の状態を再現する。単に過去の情報を検索するだけではなく、「当時行っていた作業環境をできるだけ完全に復活する」のが時間移動の目的である。利用者は、目的のファイルを見つけるだけでなく、そのとき同時に画面に登場するアイテムから当時の作業コンテキストや関連する情報をも復活させることができる。たとえば、ファイルを変更したときに、同時にデスクトップにメモ書き(ポストイット)を貼りつけておくと、後でその時刻に戻って来たときに有効な情報になる。

移動すべき日時(時刻)の指定には様々な方法が考えられる。「1999年1月1日」のように直接指定する方法のほかに、「このファイルを作成したとき」「このパラグラフを変更したとき」「このポストイットをデスクトップに貼ったとき」のように、作業履歴に依存した時間指定も可能であろう。また「XXXという文字列が画面に表示されているとき」のように検索によっても時刻を指定することができる。

時間情報の視覚化: 一方、単にコンピュータ画面の状態を復元しただけでは、時間的な遷移が把握しにくい場合がある(直前に削除されたファイルが見えない、など)。したがって、時間移動を支援する情報視覚化技法も Time-Machine Computing の研究課題となる。後述する TimeScape システムでは、半透明表示、複数ビュー(タイムラ



図 1. TimeScape デスクトップの画面例

イン・カレンダー)の提供、複数ビューのアニメーション効果による切替えなどの技法を試みている。

2 TimeScape: 時空間デスクトップ環境

Time-Machin Computing の発想に基づく、“TimeScape” とよぶ情報環境を Java によって構築中である。TimeScape は、前述の「デスクトップにすべてのファイルを置く」という作業スタイルと時間指向インタフェースの考えを組合せ、ファイルアイコンやポストイットの任意時点での状態を再現することを目的とした環境である(アプリケーション状態の再現は、後述する time-casting 機能によって部分的に実現している)。図 1 にデスクトップ画面の一例を示す。TimeScape では、フォルダ階層による分類を使わず、文書ファイルやポストイットなどのアイテムを、すべてデスクトップ画面上に直接置く。また、デジタル画像は縮小画面(サムネイル)が表示される。これらのアイテムの位置は画面上で自由に変更できるので、関連するアイテムどうしを近くに置いてグループ化したり、積み上げてパイル化することもできる。通常のデスクトップ環境と同様、アイコンをダブルクリックすると対応するアプリケーションが起動する。

2.1 時間移動機能

TimeScape の特徴は、デスクトップの状態を過去・未来に自由に移動させる機能である。画面の右上に時刻ダイヤルが表示されており、通常は現在時刻になっている。この時刻ダイヤルをマウスで操作することで、別の時刻に移動し、その時点でのデスクトップ環境が画面に再現される。また、「過去移動」「未来移動」コマンドによって、あるイベントが起きる時刻にスナップしながら時間を移動していくことができる。イベントの種類としては、アイテムの生成・削除、位置の移動がある。後述するように、検索コマンドを利用して時間移動することも可能である。また、アイテム



図 2. TimeScape デスクトップの各種視覚化技法.

を選択して、「そのアイテムが生成(消去)された時刻」に時間移動する機能や、文字列を指定して「その文字列が画面に登場するまで過去(未来)方向に移動」する機能を提供している。

時間移動を実現するために、デスクトップ環境に施された変更履歴を記録している。たとえばファイルをデスクトップから削除(通常と同様、ゴミ箱アイコンにドラッグする)しても、そのファイルの内容がコンピュータから消滅するのではなく、「削除された」という履歴情報として記録される(補助的には、履歴情報からもファイルを完全に削除する機能も提供している)。

TimeScape の、作業環境そのものを保存するという方針には、Lifestreams のように個々のファイルを時間順に整列するという方針とは、いくつかの違いがある。TimeScape のファイルは、単純に時間順に整列されているのではない。デスクトップ上では、作成の時刻に関わらず、必要なアイテム(群)を同時に保持することができる。作業内容によって、いったん画面から消去されたファイルを(過去から持って来て)再び画面に登場させることが可能である。一方、Lifestreams ではすべてのアイテムを 1次元に整列されているので、ある作業に必要なファイル群が列の各部分に分散してしまう可能性がある。TimeScape では情報を視覚的に整理できる、という従来のデスクトップメタファが持っている長所を継承しているので、複数のアクティビティを空間的なグループとして表現することができる。一方、LifeStreams ではアクティビティを表現する明確な手段は提供されていない。

TimeScape は、未来への時間移動機能も提供している。ここには「将来必要となりそうな情報」を予め置いておくことができる。この機能はスケジュール管理(リマインダ)として有効である。たとえば会議に必要な関連書類を、会議開催日に時間移動して未来のデスクトップにあらかじめ置いておくことができる。その日が訪れると、必要なファイルが自動的にデスクトップに出現する。

2.2 視覚化技法

TimeScape デスクトップでは、時間情報を表現するために幾つかの視覚化表現技法を導入している(図 2)。(a) は、貼りつけてからの経過時間によって色が変化していくポストイットの例で、情報の「新鮮度」を視覚的に表現している。(b) は、アイコ

Time-Machine Computing: Towards Time-centric User Interfaces

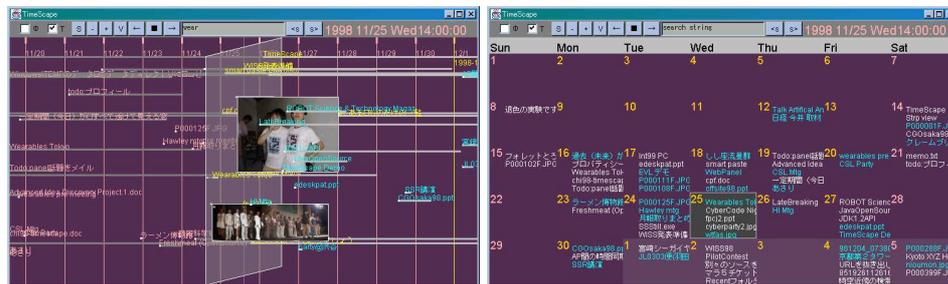


図 3. 時間情報を表現する各種ビュー：(左) タイムライン、(右) カレンダー

ンを消去した直後のデスクトップで、情報の痕跡が薄く画面に残っている(約1日後に消滅する)。ごく最近まで画面にあったアイテムを知ることができ、別の時刻に移動したときに効果的である。(c)は、アイテムの生成・消去時刻を半透明のバンドで図示している例である。画面の左端は1日のなかでデスクトップ時刻がどこにあるかを示す時刻バーになっている。そこに近づけたアイテムの生成・消去時刻が、バー上に表示される。この機能を利用すると、画面の左端付近を1日の予定管理情報として使うことができる。また、過去や未来を訪れている間、デスクトップの背景色が変化してユーザに状態の変化を通知する。

さらに、時間の経過に従って作業空間がどのように変化していくかを視覚的に把握するために、TimeScopeではデスクトップの他に幾つかのビューを提供している。タイムラインビューは、デスクトップを時間軸方向から年表風に表示する(図3左)。このビューでは、デスクトップが時間方向の「断面」として表現するために、疑似的な3次元表示を行なっている。画面中央にデスクトップに対応する長方形が斜めに表示され、左側に過去の情報が、右側に(デスクトップの時刻よりも)未来の情報が表示される。各アイテムは、生成・消滅時刻に対応した時間方向の長さを持った横方向のバーとして表示される。タイムラインのスケールをズームコマンドで連続的に変化させることができ、ユーザは毎日のアクティビティから、数年間、極端には自分の一生に渡るアクティビティの変化を連続的にブラウズすることができる。カレンダービュー(図3右)は、日毎の変化をサマライズしたセルをカレンダー状に配置したビューである。標準設定では、その日に生成されたアイテムを縮小表示している。

時間移動の際にも、複数のビューを利用することができる。たとえば、ユーザはカレンダービューの上で特定の日を選択し、「その日」に対応するデスクトップに移動し、さらにタイムラインビューに切替えて「その日の前後」に何が起きているかを知ることができる。

複数のビュー間の意味的な連続性を表現するために、アニメーションを使った画面切替えを行なっている(図4)。この視覚効果によって、たとえばデスクトップとタイムラインの(意味的な)関係を直観的に把握できるようになる。

2.3 時空間の情報検索

このように、TimeMachne Computingが扱う情報は時間方向と空間方向の次元をもつ空間となる。したがって情報検索も、従来のように空間方向(ファイル上での文字列の位置)だけではなく、時間方向に拡張することが必要である。TimeScopeでは、デスクトップ上に指定した文字列が登場するまで過去(あるいは未来)に移動する、と

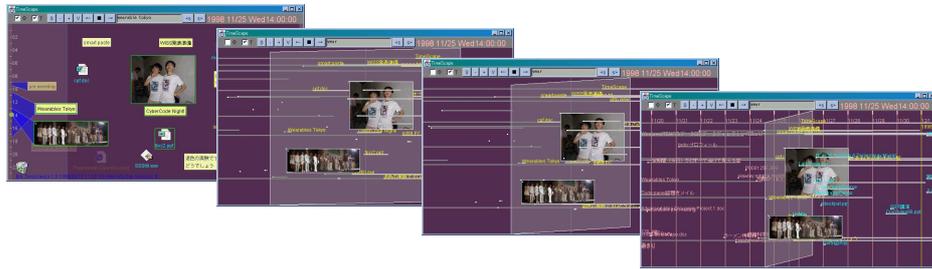


図 4. アニメーションを使ったデスクトップとタイムラインビューの切替え

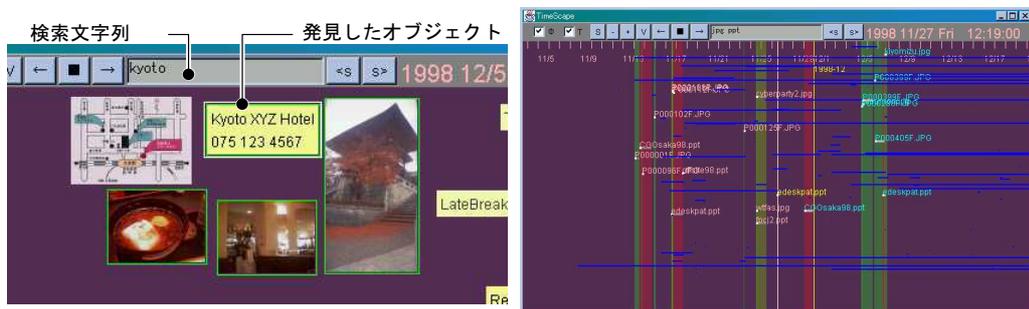


図 5. 検索機能の例：(左) 文字列“kyoto”を過去方向に検索した直後の画面。
(右) 検索文字列を含むオブジェクトを動的に絞り込む。

いう検索コマンドを提供している。

図 5(左) は文字列“kyoto”を過去方向に検索した直後の画面例である。この例では、時間移動が kyoto を含むポストイットを発見して停止している。ユーザは、単にポストイットを検索するだけでなく、当時行っていた関連する情報をデスクトップ上に見出すことができる。この例では京都旅行の写真がデスクトップに現れている。このように、デスクトップ上の文字列を、日々の情報管理 (TODO リストなど) としてだけではなく、情報を検索するための時間軸上のブックマークとして利用することができる。図 5(右) はビューをタイムラインに切り替えた状態を示している。このビューで文字列を検索ボックスに入力すると、その文字列を含むアイテムが動的にフィルタリングされ、さらに時間軸上に半透明のバンドが表示される。このバンドは、検索に合致したアイテムの個数に応じて色が濃くなっていくので、たとえば“wiss”と検索することで WISS 関連の作業が多く発生している時期をバンドの濃淡から把握することができる。

3 アプリケーション間の時刻情報による連携

TimeScope は TimeMachine Computing の考えに基づいて新規開発したシステムであるが、既存アプリケーションでも、潜在的に時間の概念をもつものは少なくない。たとえばメールブラウザは「現在表示しているメールの着信時刻」という概念があるし、写真ビューワーでも「現在表示している写真の撮影時刻」という時刻情報を持っている。またウェブブラウザではアクセスしたページの履歴情報を記録しているが、これにも時刻情報が含まれている。しかし、これらの情報は互いに連携して利用さ

Time-Machine Computing: Towards Time-centric User Interfaces

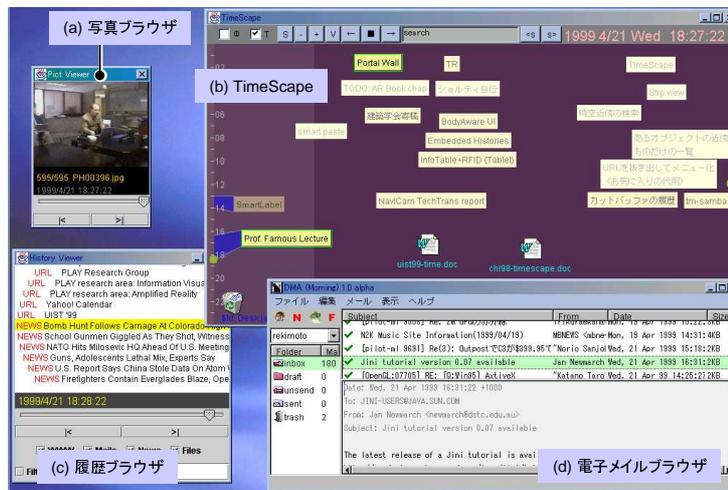


図 6. Time-casting 時間情報によるアプリケーション間連携

れることを想定していないので、「この文書ファイルを作成したところにアクセスしていた WWW サイト」、「この写真を撮影したときに参加していた会議に関する情報」といった、アプリケーションをまたいで情報を参照する場合、煩雑な操作を必要としていた。

そこで、アプリケーション間で時刻情報を通信し合って連携動作する方式を提案する。具体的には、以下のような手順を取る：

- ・各アプリケーションは「アプリケーションの時刻」という概念をもつ。たとえば画像ブラウザでは、選択した画像（写真）の撮影時刻を「アプリケーションの時刻」として定義することができる。
- ・時間連携を行うアプリケーションのグループを定義する。通常は同じ画面上に表示されているアプリケーションが同一のグループに所属するが、デスクトップと PDA というようにコンピュータをまたいでグループを形成することも可能である。
- ・ユーザ操作等によって「アプリケーションの時刻」が変化すると、アプリケーションは時刻情報をグループに所属する他のアプリケーションに通知する。
- ・時刻情報を受け取ったアプリケーションは、その時間に「アプリケーションの時刻」を合わせるように内部状態を変化させる。

このアプリケーション連携の枠組みを **time-casting** と呼ぶ。time-casting は、狭義には時間移動ではない（指定した時刻におけるアプリケーションの状態を再現しているわけでは必ずしもない）が、利用者がその時刻にどんな作業をしていたかを想起するのに有用な情報を提供することができる。

図 6 に time-casting によるアプリケーション連携の例を示す。図は写真ブラウザ (b) からユーザが 1 枚の写真を選択した直後の画面例である。TimeScape デスクトップ (a) が、自動的にデスクトップの時刻を写真撮影時に設定している。その結果、写真に撮影されている人物の名前がデスクトップのポストイット（スケジュール情報）として表示される。つまり、デスクトップ上の情報が写真のキャプションとして機能するようになる。time-casting による時間連携は双方向なので、逆にデスクトップ上で時間移動を行うと、写真ブラウザもそれに応じて、「デスクトップの時刻に最も近

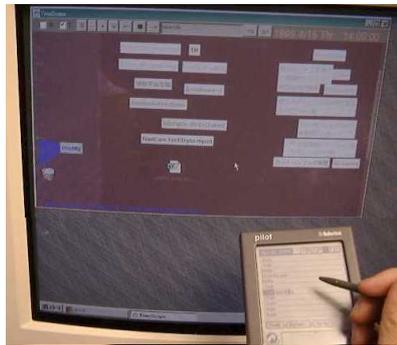


図7. デスクトップとパームトップコンピュータ間での時間連携

「い撮影時刻を持つ写真」を表示するように画面が変化し、当時の記憶をより鮮明に再現することができる。写真ブラウザと TimeScape デスクトップブラウザは独立したアプリケーションであるが、時刻情報で交換することで互いに意味を補完するように連携動作する。さらに、メールブラウザやテキストエディタを時間連携の枠組みに取り入れることが可能である。図(c)は WWW にアクセスした履歴をブラウズするアプリケーションで、指定した時間の付近にアクセスしていたリンクの一覧を表示する。(d)はメールブラウザで、メールを選択すると、その着信時刻を他のアプリケーションに通知する。

time-casting の機構は、複数台のコンピュータに適用することができる。たとえば会議室で、参加者が持ちこんだコンピュータと黒板型コンピュータが連携することで、前回の会議の情報を容易に黒板に再現することができる。図7はデスクトップコンピュータと PDA との間で時間連携を行っている例である。PDA 上でスケジュール情報やメモを選択すると、その時刻(メモの場合は作成時刻)が送信され、デスクトップ環境が対応する時刻に時間移動する。

4 システム実装

前節までに説明したシステムは(PDA との time-casting の例を除いて)すべて Java で記述されている。TimeScape はネイティブなデスクトップ環境とのドラッグ・アンド・ドロップを実現しているので、マウス操作で既存環境との情報交換が可能である。

TimeScape 内部では、すべてのアイテムは、生成時刻、消滅時刻、変更時刻の履歴、デスクトップ上での位置変化の履歴、外部リソース名(URL やファイル名)、その他の属性(ポストイットの内容など)から成っている。これらのアイテムは、生成・消滅それぞれの時刻で整列した配列で管理されており、任意時点におけるアイテムの集合とデスクトップ上での位置を迅速に再構成することができる。

ファイル内容の更新履歴を保存するために、samba サーバー [15] を改造した Tm-Samba というファイルサーバーを構築した。このサーバーは、ファイルに変更がある毎に過去の版を(利用者からは見えないディレクトリに)保存する。過去の版は時刻情報を含んだパス名記法で参照する:

```
T:\@924247906550\My Documents\wiss99.tex
T:\@1999.8.20\My Documents\wiss99.tex
```

前者は 1970 年 1 月 1 日からの経過時間(ミリ秒)で、後者は日付による指定である。

この記法を使うと、特別なバージョン管理コマンドを使わずに現在と過去のファイルを比較することができる。

5 関連研究

時間をキーにして情報を整理しようとする提案は、国内では野口の「超整理法」[17]が有名である。ほぼ同時期に Freeman らによって、コンピュータ上のファイルを時間で整列した 1 次元リスト (stream) で管理する方法 Lifestreams[3] が提案されている。Forget-me-not[7] は PDA 用の記憶補助システムであり、利用者の実世界におけるイベント (同席した人の ID や電話通信の履歴) を時間順に表示し、利用者過去の作業内容を思い出させる手がかりを与える。筆者らの Augment-able Reality システム [13] では、コンピュータを装着したユーザーに実空間に仮想的に貼つけた情報の操作を支援するが、「タイムマシンモード」に入ると過去や未来の状態を見ることができる。TimeScape はこれらの提案を発展させ、デスクトップ環境の変更履歴を蓄積し、空間的な情報管理と時間による管理を統合したものといえる。

アプリケーション内の作業履歴を記録するシステムとしては、グラフィカルエディタ Chimera[6] がある。このシステムではマクロコマンドを作成する際に編集履歴をコミックストリップ形式で表示する。Timewarp[1] は複数の作業者が同じファイルに変更を与える際に、版の更新情報をグラフ表示する。電子白板システム Flatland[12] では、白板に描かれた情報を過去の状態に遡って見ることができる。これらの研究と比較すると、Time-Machine Computing はアプリケーションの枠を越えて情報を保存している点に特徴がある。

複数の情報系列を時間によって結合しようという発想は、複数台のカメラによる撮影で利用されている (タイムコード)。またコンピュータを使ったノートテイキングシステム、たとえば [16, 14] などでは、ログ情報に時刻印を添付させて、関連する音声記録などとの連携を取れるようになっている。これらは閉じた環境内での時間連携の例だが、time-casting は、(独立に開発された) アプリケーション間でも時間による連携が可能であるという発想に基づいている。

一方、時系列情報を視覚化する研究にもいくつかの事例がある。Perspective Wall[8] は透視投影によってタイムテーブルを表示している。Spiral Calendar[9] はスケールの異なる複数のビューをスパイラル状に結合している。Dynamic Timelines[5] はズーム、半透明、3 次元表示などの視覚化技法を組み入れた写真史のブラウザである。TimeScape の視覚化表現では、複数のビューをアニメーションによって連続的に接続している。利用者がビューを積極的に切替えることで、時間上に分布した情報の構造を直観的に把握することができる。

6 結論と今後の課題

本論文では、コンピュータシステムの状態を過去や未来に移動させることを特徴とする新しいユーザインタフェース環境の概念、Time-Machine Computing を提案した。また、そのデスクトップ環境としての実現、TimeScape について述べた。

本研究の提案方式は、家庭でのコンピュータなど、通常のカテゴリや階層化では整理しにくいような情報を主に扱う領域により適していると考えている。たとえば情報家電の一種として、冷蔵庫の扉の表面がコンピュータ画面になるような場合、複雑なウィンドウやフォルダ管理よりは「何でも同じ表面に置いておけばよい」という TimeScape 的な管理がより適している。家庭での利用の場合、電子データのみなら

ず、物理的なメモ (冷蔵庫にマグネットで貼りつけたメモやチラシなど) も同じ枠組みで管理できるように拡張する必要があるだろう。一方、オフィスワークでは複数のアクティビティが並行して行われる場合がある。TimeScape 上での空間的なファイルグループによってこれらのアクティビティを表現することはある程度は可能だと考えているが、他の技法、たとえば Rooms や zooming, fish-eye ビューなどとの併用を検討している。また、PDA やページャーなどの小さな画面を持つコンピュータでは、複数のウィンドウやフォルダを操作するのは煩雑であり、本研究の単一画面方式がより適していると思われる。小型スクリーン向けの時間指向インタフェースも今後の課題である。

謝辞: システムの実装に協力して頂いた中村成貴氏に感謝する。

参考文献

- [1] W. Keith Edwards and Elizabeth D. Mynatt. Timewarp: Techniques for autonomous collaboration. In *CHI'97 Proceedings*, pp. 218–225, 1997.
- [2] Scott Fertig, Eric Freeman, and David Gelernter. "finding and reminding" reconsidered. *ACM SIGCHI Bulletin*, Vol. 28, , January 1996.
- [3] E. Freeman and D. Gelernter. Lifestreams: A storage model for personal data. *ACM SIGMOD Bulletin*, March 1996.
- [4] Frank M. Shipman III, Catherine C. Marshall, and Thomas P. Moran. Finding and using implicit structure in human-organized spatial layouts of information. In *CHI'95 Conference*, pp. 346–353, 1995.
- [5] Robin L. Kullberg. Dynamic Timelines: visualizing the history of photography. In *CHI'96 Conference Companion*, pp. 386–387, 1996.
- [6] David Kurlander and Steven Feiner. A history-based macro by example system. In *Proceedings of UIST'92*, pp. 99–106, 1992.
- [7] Mik Lamming and Mike Flynn. Forget-me-not: Intimate computing in support of human memory. In *FRIEND21 '94 International Symposium on Next Generation Human Interfaces*, 1994.
- [8] Jock D. Mackinlay, George G. Robertson, and Stuart K. Card. The perspective wall: detail and context smoothly integrated. In *Proceedings of ACM CHI '91*, pp. 173–179, 1991.
- [9] Jock D. Mackinlay, George G. Robertson, and Robert DeLine. Developing calendar visualizers for the information visualizer. In *Proceedings of UIST'94, ACM Symposium on User Interface Software and Technology*, pp. 109–118, November 1994.
- [10] Thomas W. Malone. How do people organize their desks? implications for the design of office information systems. *ACM Trans. On Office Systems*, Vol. 1, No. 1, pp. 99–112, 1983.
- [11] Richard Mander, Gitta Salomon, and Yin Yin Wong. A 'pile' metaphor for supporting casual organization of information. In *CHI'92*, pp. 627–634, 1992.
- [12] Elizabeth D. Mynatt, Takeo Igarashi, W. Keith Edwards, and Antony LaMarca. Flatland: New dimensions in office whiteboards. In *CHI'99 Proceedings*, pp. 346–353, 1999.
- [13] Jun Rekimoto, Yuji Ayatsuka, and Kazuteru Hayashi. Augment-able Reality: situated communication through physical and digital spaces. In *Proc. of the Second international symposium on wearable computers (IEEE ISWC'98)*, pp. 68–75, 1998.
- [14] Lisa J. Stifelman. Augmenting real-world objects: A paper-based audio notebook. In *CHI'96 companion*, pp. 199–200, 1996.
- [15] The Samba team. SAMBA: opening windows to a wider world. <http://www.samba.org>.
- [16] Karon Weber and Alex Poon. Marquee: a tool for real-time video logging. In *Proceedings of CHI'94*, pp. 85–94, 1994.
- [17] 野口悠紀雄. 「超」整理法. 中公新書, 1993.