

Wrapper方式に基づくハイパーメディア型文書管理システムの検討

窪野 哲光

東京電力株式会社 システム研究所情報研究室
〒230 横浜市鶴見区江ヶ崎町4-1

業務を通じて獲得された情報は事業活動を支えるかけがえのない財産であり、その財産を誰でもが有効活用出来るような情報の共有化を支援する文書管理システムが強く求められている。情報へのアクセスに注目し、Windows上の全てのアプリケーションソフトで作成したファイルを拡張属性によりオブジェクト指向ファイルシステム化し、それに対するハイパーメディア操作が出来るGUIとを連係するWrapper方式は、メディア情報とリンク情報とを陽に分離したハイパーメディアモデルである。このWrapper方式と、それに基づくハイパーメディア型文書管理システムについて検討したので報告する。

A Study on Document Management System Using Wrapper Hyper Media Model

Norimitsu KUBONO

Tokyo Electric Power CO., LTD.

Advanced Information Processing Department Computer And Communications R&D Center
4-1, Egasaki-Cho, Tsurumi-Ku, Yokohama, 230

Document created and outputed by employee through their works, which are essential information resource for Company Business Activity. Then, we study on configuration for document management system, which has original software platform "Wrapper" hyper media model which is extending MVC-Paradigm. "Wrapper" hyper media model is consist of object-oriented file system which encapsulates Windows file with user-defined-extended profile and user-friendly-interface such as hyper-link operation using only mouse and icon. This methodology separates link-information from media-information, and which sets thinking free from the conventional hyper-text system's constraints.

1. はじめに

我々の周りには多種多様な文書情報が多数あるが、必要な情報が直ぐ手元で見られない、数ある文書が有効な形で関連付けて保存されていない、保存されている文書のどこが重要な箇所かが明示されていない等の理由により、時間が経過すると、保存されている文書が別の人に有効に活用されず、折角のデータが死蔵されているのが現状である。

社員の頭脳が生み出した情報・業務を通じて獲得された情報は、事業活動を支えるかけがえのない財産であり、このような問題点を解決するため、多種多様な文書情報と関連情報とを、それがネットワーク上のどこにどのような形式で格納されていても、素早く簡単に検索し活用できるようなサービスを提供する文書管理システムの構築をコンセプトとして検討を進めており、スタンドアロン型、及び、マルチサーバ型への対応が出来るデータベースシステムの2形態のハイパーメディア型文書管理システムを開発した。

その構成法であるWrapper方式は、情報へのアクセスに注目し、Windows上の全てのアプリケーションソフトで作成したファイルを拡張属性によりオブジェクト指向ファイルシステム化し、それに対するハイパーメディア操作が出来るGUIとを連係する方式であり、メディア情報とリンク情報とを陽に分離したハイパーメディアモデルである。

2. 文書管理に於ける現状の問題点

2.1 Windows ファイルシステム

ファイル名の文字数の制限などで個人情報管理でさえ苦勞しているのが現状であり、ネットワーク環境で複数の人間が文書ファイルを共有する場合にはなお更である。このため、普通は複数のディレクトリに分けて分類するが、この場合でも誤ったディレクトリに入れてしまう危険性がある。

そもそも、分類というのは一意に決まるもので

はなく、様々な見方に応じた分類が存在するのである。また、文書管理に於いては、予め文書の分類体系を決定出来るものではなく、文書の量が溜まった段階で分類方法が決まるというものが多く、また、利用者のノウハウの蓄積度により分類体系が進化していくものでもあり、これをサポートする必要がある。

2.2 ハイパーテキストシステム

ハイパーテキストを利用した情報整理は確かに有効ではあるが、一般のハイパーテキストシステムでは以下の制約がある。

- ・ 1 : 1 リンクだけであり、リンクの表現力に乏しい。

- ・ リンク付け操作はスクリプトを書かなくてはならないため、プログラム作成には不慣れな一般の人には日常業務の中では容易には使えない。

- ・ ハイパーリンクはパス名等の物理的な情報を用いた参照方式であるため、文書の分類階層を変更するとハイパーリンク情報に不整合が発生する。

つまり、組織情報管理を指向し、日常業務で利用する市販OAソフトで作成した多種多様なマルチメディア文書情報のシームレスなリンクの作成、一貫性のある維持管理をサポートする必要がある。

2.3 オブジェクト指向システム開発

オブジェクト指向技術でシステム化する以前に、Windows上の多彩なアプリケーションで作成した多種多様な文書データが既にあり、また、種類も増えていくが、それらをどのようにオブジェクト化して取り扱うかが問題である。

つまり、既存のWindowsファイルシステムの利用形態を包含しつつ、情報の格納・情報の交換・情報の付加価値化をより効果的に行なうために、多種多様なマルチメディア情報を多種多様なまま扱うのではなく、規格化された同じ属性を持つオブジェクトでカプセル化して抽象的に扱うことが必要となる。

3. ハイパーメディア型文書管理システムの構成

3.1 設計指針

MVCモデルの拡張に基づく独自のモデルを枠組みとし、オブジェクト指向ファイルシステムとハイパーメディア型ユーザインタフェースを組み合わせた構成法を取る。

米国ゼロックス社で開発されたオブジェクト指向言語Smalltalkの開発環境であるVisualWorks\ Smalltalkを利用して開発を行った。

3.2 オブジェクト指向ファイルシステム

3.2.1 Windowsファイルのカプセル化

多種多様なマルチメディア情報を多種多様なま扱うのではなく、情報の格納・交換・付加価値化をより効果的に行なうためには、情報に機能を付加し、統一的な操作が出来るように、Windows上の全ての文書ファイルを、規格化された同じ属性を持つオブジェクトでカプセル化して抽象的に扱うことで(Wrapper化)、文書ファイルの中身の差異を隠蔽して抽象化を高めることが出来、利用者には操作方法の統一化、プログラマにはAPIの標準化を提供出来る。

文書ファイル本体の中にパス名などの静的リンク情報を直接に埋め込む方式は、文書管理の観点からは大きな問題があるが、Wrapper化により文書ファイル本体とは別にリンク情報を格納・管理することで、自由度の高い豊富なセマンティックスを持つ動的リンクを実現出来る。

Wrapper属性には、リンク情報以外には、キーワード、文書名(Windowsのファイル名とは別に利用者が与える文書名)等の属性があり、増やすことは可能である。

3.2.2 メディアに対する拡張属性・付加知識

Wrapperはメディアに対する拡張属性と

して捉えることが出来る。文書情報の共有化を図るためには、単にデータという素材だけを対象とするだけでは不十分であり、そのデータを評価した情報・知識をも表現し共有化する必要がある。そもそも、データとデータとが結びつくときに情報は発生し、データの見方や役割は結び付きが決定することを考えると、ハイパーリンク情報も、この拡張属性の一つとして捉えるべきであり、情報の共有化のためには、知っているということを明示的に示すことが有効であり、各人のノウハウを文書同士のリンク付けという形で蓄積・継承することで促進される。

3.3 ハイパーメディア型ユーザインタフェース

3.3.1 ハイパーメディアによる文書管理の有効性

文書資料を整然と整理しなくても、本人の頭の中には、文書の所在、文書間の関連性は出来上がっている。誰でもが頭の中に自分なりの「情報の空間」を持っており、その情報空間を泳ぎ回って、目指す情報を次々にピックアップしながら仕事を進めている。つまり、情報同士の間には目に見えない連想のリンクが張られており、これをハイパーリンクで表現し管理することが有効である。

情報空間とは、現象のモデル化としての理解の構造を表現したものであり、我々が理解をするということは、既に持っているモデルを通して現象を理解することである。新しい情報に遭遇した際には、既存のモデルの一部を修正・拡張するという形で、その中に埋め込まれていく。(図1)

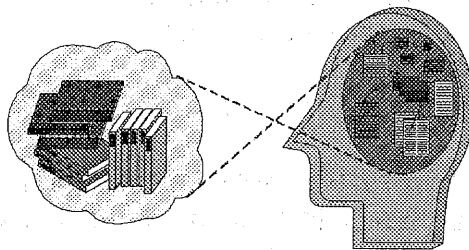


図1. 情報の空間

オブジェクト指向の概念をハイパーテキストモデル上に展開し、多種多様な文書をオブジェクトとして扱い、オブジェクト間にメッセージ交換で実現されるリンク（1：N）が付与される。また、このリンクには属性（リンク属性）が付与出来るので、通常のハイパーテキストモデルに於けるリンクの概念より豊富なセマンティックスを持つ。

3.4 MVCモデルの拡張に基づく枠組み

3.4.1 MVCモデルの概要

MVCモデルは対話型グラフィックスシステムを構築する際の規範的モデルとしてSmalltalk-80で提案されたものであるが、Smalltalk 以外のオブジェクト指向システム開発でも広く利用されている。MVCモデルは、Model-View-Controllerの3つ組のオブジェクトから構成される仮想型複合オブジェクトであり、以下の役割を持つ。

- ・Modelはアプリケーションの扱うデータとそのデータに係わる操作を担当する。

- ・ViewはModelを画面に表示する手段を提供する役割を持ち、表示対象となるModelを画面上のどの領域に書くのか、どうやって表示するのかを担当する。

- ・Controllerは、ユーザーからの入力（キーボード、マウス）を解釈して、ModelやViewに適切なメッセージを送ることを担当する。

MVCモデルは、問題領域の構造を表現するアプリケーション固有部と、図形表示インタフェースに係わる処理部とを分離することの有効性を示し、アプリケーション全体の開発効率の向上とインタフェースの統一に大きく貢献した。

3.4.2 MVCモデルの拡張

対象システムがマルチメディア的な性質を有する場合、画面上の構成要素・ユーザー拡張属性などをModelに含めたMVCでモデリングすると幾つかの大きな問題が発生する。

そこで、文書ファイル本体とユーザー拡張属性（Wrapper）の2つのモデルを独立・明示的に表現し、2つのモデルを協調させる枠組みが必要になる。つまり、Wrapperを独立したオブジェクトとし、MVCにおいてModelと同等に捉えることを可能にする枠組みの導入が必要となる。

MVCモデルの拡張型アプリケーション構築枠組みとして、MVCモデルに4番目のオブジェクト（Wrapper）を導入し、Wrapper-Model-View-Controllerの4つのオブジェクトから構成される。

ViewとWrapperとの関係は、そのままViewとModelとの関係に置き換えることが出来る。つまり、View/ControllerからWrapperへの関係付けは、Wrapperオブジェクトを自身のインスタンス変数として明示的に保持し、WrapperからViewへの関係はdependencyの概念を用いて非明示的な参照関係を与える。

- ①ユーザー拡張属性を操作する場合には、それをモデル（Wrapper）として捉える。

- ②文書を操作する場合には、文書ファイル本体をモデル（Model）として捉える。

- ③Wrapperを通して、多種多様な文書ファイルを統一的に操作するという3つの直交した観点でアプリケーションを捉えることが可能となる。

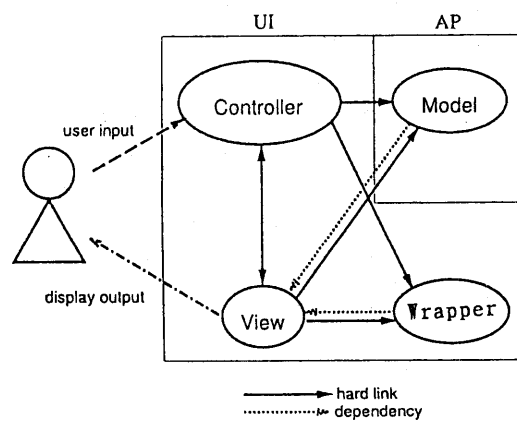


図2. MVCモデルの拡張の概念図

4. 本システムの機能

4.1 文書管理のための共有画面

利用者側から見た文書管理のための共有画面である「文書キャビネット」を提供する。Windows ファイルマネージャから文書キャビネットへ、ファイルアイコンを drag & drop 操作することにより、本システムへデータ登録がされる。

文書ファイル本体と拡張属性 Wrapper との結び付けは、登録時点で間接実行されるので、利用者が操作する必要はない。以降、文書へのアクセスは Wrapper を通して行うことになる。登録された文書に対する、個人毎・視点毎のアクセス画面も設定出来るので、組織情報と個人情報との共存が図れる。

4.2 視覚的な操作による動的リンク付け

スクリプト記述・マークアップ等の煩わしい操作を一切必要としない、マウスとアイコンの視覚的な操作で簡単に動的リンクが作成出来ることにより、「情報の共有化」を指向して誰でもがリンク作成者として参加出来るようになった。

(1) ラバーバンド方式

リンク元文書にリンクボタンを作成し、リンクボタンからリンク紐をラバーバンドで引き延ばし、

リンク先文書のリンクアイコン上でクリックする。

(2) アイコン方式

リンク先文書のリンクアイコンをドラッグしてリンク元文書にドロップする。

例えば、テキスト文書は、リンク付けされた状態での文書編集可能な機能を有する。

4.3 文書のリンク付け状態の視覚化

本システムに登録された全ての文書ファイルの参照～被参照状態のグラフ図を自動作成する。リンクの多重度は重要な文書の目安となる。また、単にグラフ図を作成・表示するだけでなく、グラフ図から文書へのアクセス、グラフ図上で文書同士のリンク付けも出来る。(図3)

4.4 文書キャビネットの動的な分類階層

文書管理に於いては、予め文書の分類階層を全て決定出来るものではなく、文書の量が溜まった段階で分類方法が決まるというものが多く、また、利用者のノウハウの蓄積度により分類階層が進化していく。文書キャビネットに登録された文書については、文書の分類階層を変更しても、設定したリンク付けは保存されるので、予め分類階層を定められない文書の体系的な管理を柔軟性良く行うことが出来る。(図3)

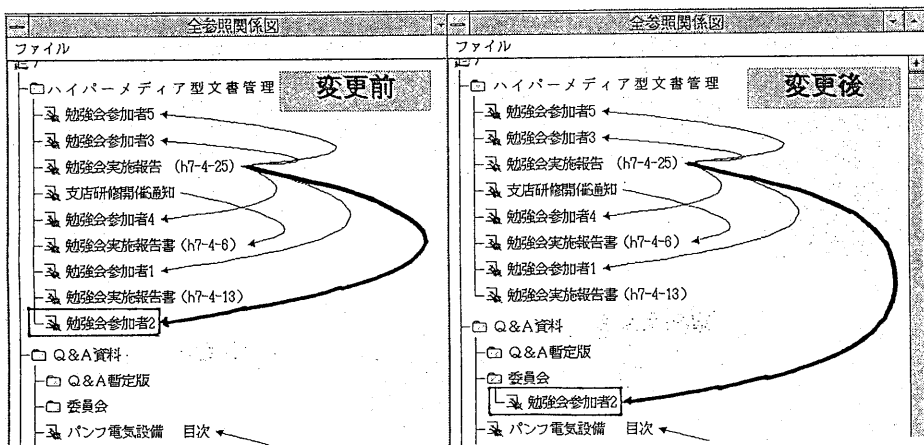


図3. 文書のリンク付けの視覚化 と リンク付けを保持した状態での動的な分類階層

4.5 文書検索

4.5.1 ハイパーリンクによる文書検索

ハイパーリンクによる連鎖的な検索では、最初のアクセスポイントを見つけるのが大変である。そこで、

- ①分類階層（文書キャビネット、リンク図）
- ②時間（年表、カレンダー、タスク図）
- ③人（個人情報、名刺情報、組織図）

等の多面的な視点からの文書管理・検索が出来る。（図4）

4.5.2 文書型の指定による検索

文書にオブジェクト型を導入することで、文書の種類毎にアイコン化表示出来るため、対象とする文書のアイコンを選択するだけで検索をすることが出来る。文書名の中の特定の語句、付与したキーワード等の条件設定をすることもできる。検索結果を用いた一括リンク付けにより、抜けが無

く効率良いリンク付けが出来る。

4.6 不定形文書のデータ構造最適化方式

オブジェクト指向システム開発では、クラスライブラリ化・インヘリタンスによる差分プログラミングが大きな特徴であるが、データ構造は無条件に継承されてしまうので、注意深い設計が必要となる（一方、メソッドはオーバーローディング等で対応出来る）。特に、マルチメディア情報を持つ不定形な文書に対しては予めデータ構造を決めることが出来ないため、データ構造を如何にフレキシビリティに保持するかが重要となる。

そこで、Smalltalk システムが提供するクラス階層に基づくインスタンス変数の継承ではなく、メタクラスに自分達が与えたインスタンス変数の継承を使うことで、個々のクラス毎に同じ名前で異なる領域をクラス変数で持つことが出来るようになり、文書毎に最適なデータ構造を与えることが出来る。

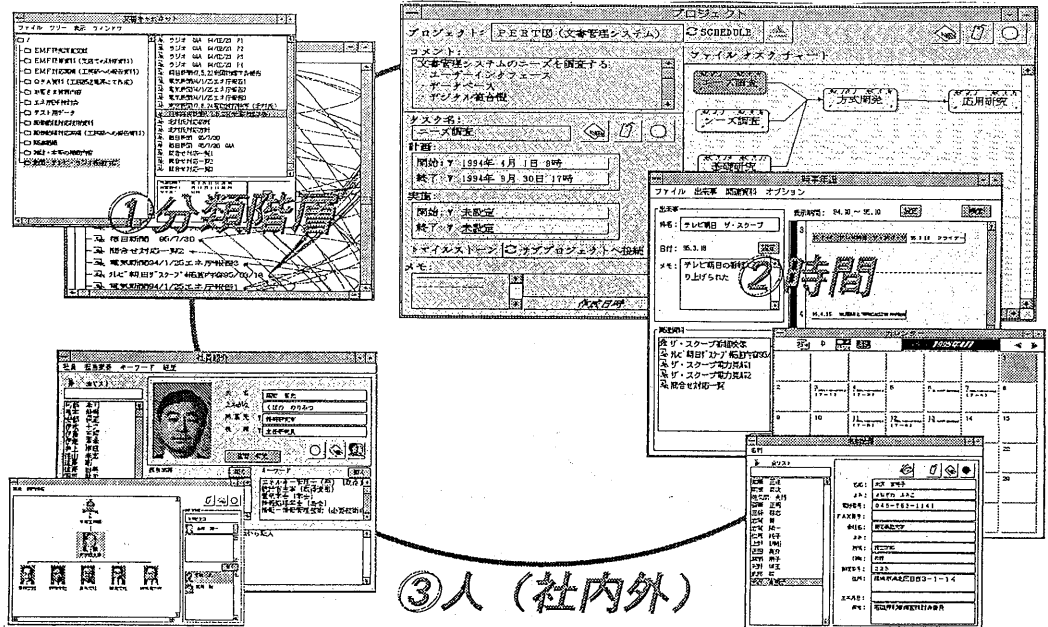


図4. 分類・時間・人などの多面的な視点からの管理、検索

5. 文書管理データベースシステム

スタンドアロン型の構成法の利点を活かした形態で、マルチサーバ型（分散DB型）への対応が出来る、3階層クライアント／サーバ方式を適用した文書管理データベースシステムを開発した。

5.1 プレゼンテーション層：ハイパーメディア型GUI

スタンドアロン型の文書ファイル本体管理部・拡張属性管理部をサーバ側に移して代行させることでアプリケーションモデル部を軽くした、ハイパーメディア型GUIアプリケーションである。

5.2 アプリケーション層：文書管理ミドルウェア

ミドルウェアを選択する場合には標準化の動向に反していないかという点を考慮する必要がある。文書管理のための標準化としてDMA（Document Management Alliance）が組織されており、それに準拠するプロトタイプの評価をするためにテスト的に利用した。

- ・文書管理部：アプリケーションモデルの定義
- ・ユーザー管理部：アクセス権の一元管理
- ・サービス管理部：付加価値サービスアプリケーション
- ・外部アクセス管理部：マルチサーバ化（次期版）

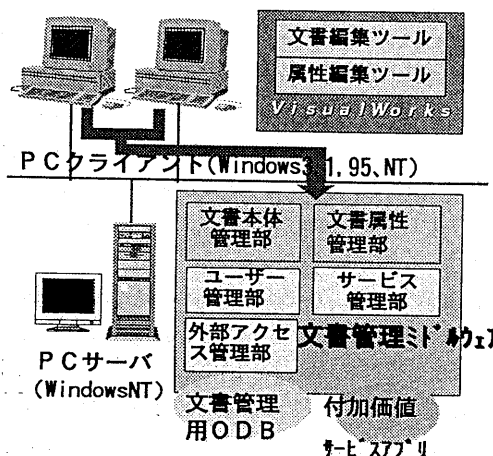
5.3 データ層：オブジェクト指向データベース

文字・数値型DBではなく、長大なマルチメディア文書、可変長・構造を持つ属性情報を直接に格納し検索する文書DBのためには、オブジェクト指向データベースが適している。更に、アプリケーションに適した部品を選択・組合せにより、文書管理に最適なストレージシステムを構築出来るという特徴を有するコンポーネント型ODBSを利用した。

5.4 文書管理ミドルウェアの効果

PC上の多彩なアプリケーションで作成したアルチメディア文書情報・拡張属性をサーバ側で管理することにより、ネットワークを介して多数での文書情報の共有化が出来る。

文書管理ミドルウェアが未だマルチサーバ化されていないので確認は出来ていないが、ミドルウェアが提供するUID（Universal ID；分散DB環境下のオブジェクト識別子）とWrapper構成法の利点を組み合わせることで、マルチサーバ間で分散された文書同士のリンク付け、および、各サーバでの分類階層が変更されてもリンク付けは保存されるという、ネットワーク型リンクを実現出来るという見通しを得た。



・プレゼンテーション層【GUI層】
スタンドアロン型のGUI部を独立

・アプリケーション層【業務モデル層】
文書管理ミドルウェア

- ・文書本体/属性管理部：クライアント側の負荷軽減
- ・ユーザー管理部：アクセス権の一元管理
- ・サービス管理部：付加価値サービスアプリケーション登録
- ・外部アクセス管理部：システム拡張性

・データ層【データ管理層】
文書管理オブジェクト指向データベース

図5. 文書管理データベースシステム（クライアント／サーバ型）

6. Wrapper方式を用いた関連研究

6.1 制約指向プログラミング

グラフィックスアプリケーションでは重要な図形のエコー表示処理などを、Wrapperにより実現された制約ソルバが、オブジェクト同士の依存関係による複雑な処理などを自動的に行う。

6.2 クラスライブラリの融合化

UIのクラスライブラリとODBMSのクラスライブラリを多重継承することにより、この新しく定義したクラスはUIとODBMSの2つの性質と機能を保有し、UIのViewからも認識出来、またODBMSのクラスとしてデータベースへのアクセスも可能となる。

7. アプリケーション事例

多種多様な文書情報と関連情報の迅速・的確な検索、活用が求められる業務での活用を検証するために、スタッフ部門での事務文書、流通設備部門での技術文書の管理を対象として試験実施を進めている。

8. 課題、今後の計画

8.1 Dexterモデルとの比較検討

幅広いハイパーテキストシステムにおける標準的なモデルとしてDexterハイパーテキスト参照モデルが提案されており、その実装方式の観点から、Wrapper方式の評価検討を行う。

8.2 Wrapper方式への推論機能の組み込み

Wrapperをメディアに対する特徴記述と捉えて推論機構を組み込み、動的な推論によってハイパーリンクを生成する等の問題解決力を向上

させる。

8.3 ブラウジング、文書情報の可視化

データ量が膨大になった時に、所望する文書への検索コストを低減するための検索方式としてのブラウザ、情報可視化ツールが必要である。

参考文献

- [1]西垣：マルチメディア, 岩波, 1994
- [2]春木：Smalltalk-80による対話型アプリケーションの構築、チューリングマシン、Vol. 2、No. 6、1989
- [3]増田、笠原：Smalltalk-80における拡張MVCモデルとその応用、情報処理学会論文誌、Vol. 31、No. 2、1990
- [4]桑原、藤村、富田：MVCの拡張、それに基づくUIMS、Talkieの実装、コンピュータグラフィクス、Vol. 9、No. 1、1992
- [5]ハイパーメディア特集、Communications of the ACM, Vol. 37, No. 2, 1994
- [6]Patric Dark etc. : Integrated support for complex objects in a distributed multimedia design environment, JOOP, January, 1994
- [7]田中：ハイパーメディアとその展開、テレビジョン学会誌、Vol. 45, No. 8, 1991
- [8]銭、谷崎、原、田中：ハイパーテキストデータベースシステムTextLink/Gemにおけるオブジェクトとスキーマの動的、段階的な構築、信学技法、DE92-39, 1993
- [9]大本、田中：伝搬ビュー、オブジェクト指向データベースにおける半順序関係に基づくインスタンス間継承機構、情報処理学会論文誌、Vol. 34, No. 8、1993
- [10]W.Lalonde, J.Pugh : Subclass ≠ Subtype ≠ is_a, JOOP, Vol. 3, No. 2, 1991
- [11]P.Wegner : Concepts and Paradigm of Object-Oriented Programming, ACM OOPS MESSENGER, Vol. 1, No. 1, 1990
- [12]S.Danforth, C.Tomlinson : Type Theories and Object-Oriented Programming , ACM Computing Surveys, Vol. 20, No. 1, 1988