

感性情報処理アルゴリズムの管理と感性検索システムの自動生成への応用 —半構造データモデルを用いたアルゴリズムとシステムの一元的な管理—

荻野 晃大^{†1} 加藤 俊一^{†1}

本論文では、感性情報処理アルゴリズムや感性検索システムをデータベースにより動的に管理する仕組みと、この感性情報処理アルゴリズムを組み合わせることにより、感性検索システムを動的に生成するための仕組みを提案する。我々は、データの種類・型・数がそれぞれ異なる感性情報処理アルゴリズムと、アルゴリズムの種類と数がそれぞれ異なる感性検索システムを半構造データモデルにより統一的に記述する方法を定義し、それらをデータベースにより管理する機構を開発した。我々は、これを使って感性情報処理アルゴリズムを組み合わせ、感性検索システムを動的に合成する機構とその操作言語を設計・開発した。我々は、この2つの機構と操作言語を実装したプラットフォームを開発し、個人、コンテンツ、検索の仕方に対応させて、視覚的な類似性や言語的な解釈による感性などによる検索に対応した、感性検索システムの動的な生成することを可能とした。

Management of KANSEI Information Processing Algorithms for Automatic Synthesis of KANSEI Information Retrieval Systems Based on Semi-structured Data Model

Akihiro OGINO,^{†1} Toshikazu KATO^{†1}

We proposed a management method of KANSEI information processing algorithms for automatic synthesis of KANSEI information retrieval system. We designed a management language for designing KANSEI retrieval system by combining algorithms in a simple manner. System developers have only to define algorithms and systems on semi-structured data model to develop KANSEI retrieval systems. System developers choose and combine algorithms, and can develop information retrieval systems on our platform. We have also briefly estimated our platform.

1. はじめに

人々が実世界の様々な事物に接した時に、その事物を知覚し、類型化し、概念の印象や主観的な評価と結びつける過程は、個人ごとに異なる。本研究ではこの過程のことを「感性」と呼ぶ。

現在、実世界の事物を表す画像や VRML 等(以後、コンテンツ)に対する感性をモデル化し、個人の感性に適したコンテンツをデータベースから検索する、感性検索システムの研究・開発されている²⁾⁻⁴⁾。

従来の感性検索システムに関する研究は、特定の人やコンテンツに対する感性をモデル化しており、そのシステムの開発は個別的に行われてきた。

このような個別的な感性のモデル化とそのシステム開発では、色々な人、コンテンツ、検索の仕方に対応できないため、情報サービス、ヒューマンコミュニケ

ーション支援などに感性のモデル化の技術を応用・利用することが難しい。このような問題から、汎用性のある感性のモデル化とそのシステムの確立が急務となっている。

この問題に対して我々は、従来の研究で生成されたり、新規に作成されたりした、コンテンツの知覚や類型化などの感性情報処理アルゴリズム(以後、アルゴリズム)を一元管理し、それらを適時組み合わせ、視覚的な類似性に関する感性や言語的な解釈による感性などによる検索を行う感性検索システム(以後、システム)を設計・利用・蓄積することによる解決を試みる。

本研究では、以下の仕組みとその操作言語をもつソフトウェアプラットフォームを開発した。

- ・アルゴリズム(感性を模倣する情報処理)とシステム(アルゴリズムを組み合わせたマクロに相当)を、データベースを利用して一元管理する

- ・システムをアルゴリズムの組み合わせにより、動的に生成する

^{†1} 中央大学理工学部
Faculty of Science and Engineering, Chuo University

本研究では、このプラットフォームを用いて各個人の要求に基づいた視覚的な類似性や言語的な解釈による感性検索などを行う感性検索システムの動的な生成とその実行を実現した。

本論文では、アルゴリズムとシステムを、データベースを用いて管理する仕組みと、これらアルゴリズムを組み合わせることによりシステムを動的に生成する仕組みを提案し、この2つの機能とその操作言語を持ったソフトウェアプラットフォームの有効性を示す。

2. 感性のモデル化

我々は、感性は4つのレベルから階層的に構成されていると考え、提案してきた[1].

(1) 物理レベル(Physical Level)

物理レベルでは、実世界の事物から発せられる信号を、感覚器が物理的・化学的な刺激として受容し、神経回路上の電気信号として変換する。

物理レベルをモデル化することにより、写真の構図をラフに描いて、これをキーとして提示して、同じ構図の写真を検索することができる。

(2) 生理レベル(Psychological Level)

生理レベルでは、初期視覚(明るさや色彩の平均やコントラスト等)などの感覚器の生理的な応答特性や種々の特徴抽出の神経回路に基づいてコンテンツを受容する。

生理レベルをモデル化することにより、ラフに描いた図形や手書きの文字をキーとして提示して、オリジナルのデザインの図形や印刷用の文字フォントを検索することができる。

(3) 心理レベル(Psychological Level)

心理レベルでは、生理レベルの解釈に各個人の知識などにもとづいた重み付けを行い、各個人の基準でコンテンツを類別したり、コンテンツ間の類似度を与えたりして、コンテンツを解釈する。

心理レベルをモデル化することにより、ある図形やデザインをキーとして提示して、各個人にとってキーとよく似た印象を与える図形やデザインを検索することができる。

(4) 認知レベル(Cognitive Level)

認知レベルでは、生理や心理レベルにおける解釈を、心象を表す言葉(例えば、“シンプル”など)に対応付けて解釈する。

認知レベルをモデル化することにより、各個人がそれぞれ画像に対してイメージ・表現した言葉をキーとして提示して、その人のイメージを表現した言葉にふ

表 1. 感性情報処理アルゴリズムの種類

Table 1. Kind of KANSEI Information Processing Algorithm

アルゴリズムの種類	説明
分析アルゴリズム	感性の各レベルの特徴を分析・数量化し、特徴空間を生成する
モデルアルゴリズム	各レベルで数量化された特徴を個人の感性に基づいて上位のレベルへ写像する(各特徴で構成される多次元ベクトル空間の生成)
検索・規定アルゴリズム	各個人の感性に基づいて写像された特徴空間内を検索や、その空間を規定する

さわしい画像を検索することができる。

各レベルに対応したアルゴリズムは、様々な種類が開発されている。心理や認知レベルなどの階層が上位に対応した感性検索は、下位のレベルのアルゴリズムを利用して処理を行っている。

したがって、一般性を持った感性検索を実現するためには、それぞれの感性検索の枠組みに合わせて、各層のそれらアルゴリズムを組み合わせるシステムとして設計する必要がある。

3. アルゴリズムとシステムのデータベースによる管理

既存の感性検索システムは、感性レベルごとにそれぞれのアルゴリズムを用いて、個別的に設計・構築されており、それぞれ限定的な感性検索しかできない。

このような問題を本研究では、表1の3種類のアルゴリズムとそれらを組み合わせるシステムデータベースを用いて管理し、適時利用するという方法で解決する。

アルゴリズムとシステムを、データベースを用いて管理するためには、

- (1) 感性検索ごとに、コンテンツの種類・型、教示方法のデータ表示方法が異なる
- (2) アルゴリズムごとにデータの種類(教示データ、コンテンツ)・型(file, string, binary, etc)・数(引数)が異なる
- (3) システムごとに、利用する感性情報処理アルゴリズムの種類と数が異なる

という問題を解決し、これらを統一的に表現・管理する必要がある。

本論文では、この技術的な問題の(2), (3)に関する解決法について述べる。(1)の解決法については、文献[9]で述べている。

この(2),(3)を解決する方法として、本研究ではアルゴリズムとシステムを半構造データモデルで表現し、データベースを用いて管理する。

3.1 半構造データモデルの利用

半構造データモデルは、不規則でデータ構造が一意に決まらないようなデータ群を固定的なスキーマとして構造化して管理するのではなく、そのデータの構造を自己記述的に表現することができる[5].

本研究では、半構造データモデルの一例として、関係データベースにマッピングしやすい OEM(Object Exchange Model) [5],[6]を用いて、アルゴリズムとシステムの表現と管理を行った.

本研究では、アルゴリズムとシステムの管理に必要なデータ型として、OEM のデータ型に、プログラムの種類とコンテンツのデータ型を追加した.

【OEM の定義】

Object ::= < Object-ID, Label, Type, Value >

- Object-ID : 半構造オブジェクトのための可変長でユニークな識別子.
- Label : オブジェクトの概念・意味を記述する可変長の文字列.
- Type : オブジェクトのデータタイプ.
 - 原子型: integer, string, etc.
 - プログラムの種類 : Java, C, PHP, etc
 - 画像のデータ型 : ppm, bmp, jpg, etc
 - これらデータタイプの集合または一覧表
- Value : 可変長値またはオブジェクト識別子

半構造データモデルは、リレーショナルデータモデルのように、スキーマ情報をあらかじめ定義しない。スキーマ情報は、オブジェクトの Label により定義される。本研究では、アルゴリズムとシステムを管理するための属性ラベルとして表 2 を定義した。

3.2 OEM を用いたアルゴリズムとシステムの表現

アルゴリズム M1(物理レベルのモデル化)を C や Java 等コーディングしたものを OEM で表現した例が図 1, コーディングされたアルゴリズムを組み合わせることにより生成されるシステム A1(心理レベルの感性検索)を OEM で表現した例が図 3 である。

OEM と表 2 の属性ラベルを用いることにより、感性検索システム開発者(以後、開発者)は、感性のレベル、コーディング言語、データの種類・型・数が異なるアルゴリズムを、感性のレベルによりアルゴリズムの種類・数が異なるシステムをそれぞれ表現することができる。

表 2. アルゴリズム・システムを表現するための属性ラベル

Table 2. Attribute Label for Algorithm and System

ラベル	説明
SIMULATE	アルゴリズム・システムが対応している感性のレベル
INPUT	アルゴリズムプログラムの入力を表す
OUTPUT	アルゴリズムプログラムの出力を表す
PATH	アルゴリズムプログラム本体へのパスを表す

```
/* 画像カラーヒストグラム算出*/
分析アルゴリズムM1::=
<Id1, M1 Set, {Id1-1, Id1-2, Id1-3, Id1-4, Id1-5}>
/* このオブジェクトが表現しているもの */
<Id1-1, KIND, String, "Method">
/* メソッドが模倣する感性のレベル */
<Id1-2, SIMULATE, String, "物理レベル(Physical)">
/* メソッドの本体へのパス情報 */
<Id1-3, M1, C, /method/M1 >
/* 入力1 */
<Id1-4, INPUT, ppm, "PPM形式の画像">
/* 出力1 */
<A1-4, OUTPUT, file, "カンマ区切り(Double型)">
```

図 1. アルゴリズム M1 の OEM 表現例
Fig 1. Example of Algorithm by OEM

```
/* 心理レベルの感性検索システム(一例)*/
システムA1::=
/* 管理している処理工程の統合情報 */
<Id4, A1, Set, {Id4-1, Id4-2, Id4-3, Id4-4, Id4-5}>
/* このオブジェクトが表現しているもの */
<Id4-1, KIND, String, "Algorithm">
/* 模倣している感性 */
<Id4-2, SIMULATE, String, "心理レベル(Psychological)">
/* 第1処理 */
<Id4-3, M6(色ヒストグラム+明暗特徴), set, {Id4-3-1, Id4-3-2}>
<Id4-3-1, PATH, C, /method/M6>
<Id4-3-2, INPUT, File, Input(Bmp画像)>
/* 第2処理 */
<Id4-4, M2(判別分析), set, {Id4-4-1, Id4-4-2, Id4-4-3}>
<Id4-4-1, PATH, C, /method/M2>
<Id4-4-2, INPUT, File, #(Id4-1の処理結果を入力)>
<Id4-4-3, INPUT, File, Input(教示データ(Int型))>
/* 第3処理 */
<Id4-5, M8(ユークリッド距離計算), set, {Id4-5-1, Id4-5-2}>
<Id4-5-1, PATH, C, /method/M8>
<Id4-5-2, INPUT, File, #(Id4-1の処理結果を入力)>
```

図 2. システム A1 の OEM 表現例
Fig 2. Example of System by OEM

3.3 関係データベースへのマッピング

本研究では、OEM で表現したアルゴリズムとシステムを、モデル写像アプローチ[7]による関係データベース(RDB)へのマッピングを行うことでデータベース化した[8].

本研究では、OEM 表現されたアルゴリズムとシステムを、OID, LABEL, TYPE, VALUE の属性をもつ関係表に変換して、RDB にマッピングした(図 3).

oid	label	type	value
1	PATH	c	M1
2	INPUT	pow	Binary:zpon
3	OUTPUT	file	ConnPhase:double
4	SIMULATE	String	Physiology
5	KIND	String	ALGORITHM
6	M1	SET	[[[4]], [[2]], [[8]], [[4]], [[5]]]

11	SIMULATE	String	Physiology
12	KIND	String	ALGORITHM
13	M2	SET	[[[7]], [[8]], [[9]], [[10]], [[11]], [[12]]]
14	PATH	c	M2
15	INPUT	file	ConnPhase:double
16	INPUT	file	outputfile
17	OUTPUT	file	outputfile
18	MS:POA	SET	[[[14]], [[15]], [[16]], [[17]]]
19	PATH	c	M4
20	INPUT	file	1
21	INPUT	file	1

図 3. アルゴリズムの RDB へのマッピング例
Fig 3. Example of Mapping Algorithm to RDB

データベースでアルゴリズムとシステムを管理したことにより、我々はアルゴリズムの追加や更新、システムの追加や更新を可能とした。

4. ソフトウェアプラットフォームの構築と利用

半構造データモデルを用いることで我々は、アルゴリズムとシステムのデータベースを用いて管理する機構を実現した。我々は、この機構とともにアルゴリズムの生成とシステムの動的な合成・実行を行うための機構とその操作言語を開発し、ソフトウェアプラットフォームとして実装した。

4.1 ソフトウェアプラットフォーム概要

本プラットフォームは、操作言語の解析部、アルゴリズムの登録部、システムの生成・実行部である情報処理機構(Java)と、生成・設計したアルゴリズムとシステムを蓄積するデータベース機構(PostgreSQL)をLinux(Turbo 10)上に構成した(図 4)。また、本研究で登録したアルゴリズムと生成したシステムを、表 4,5 にそれぞれ示した。また、システム実行例を図 5,6 に示した。

4.2 操作言語の定義

操作言語は、以下のように定義した。操作言語のコマンドとその機能をまとめたものが表 3 である。

【操作言語の定義】

<操作言語> ::=

<コマンド> <名前ラベル> <付加情報> “<式>”
 <コマンド> ::= CREATE, INSERT, EXECUTE,
 TEST

<名前ラベル> ::= アルゴリズム・システムを表す任意の文字列

<付加情報> ::= <ラベル=任意の文字列> |

<ラベル=任意の文字列>, <付加情報>

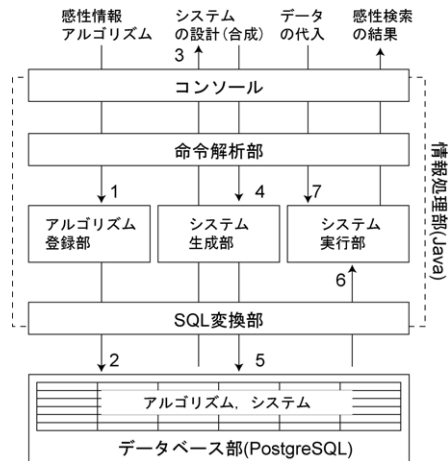


図 4. システム概要図
Fig 4. System Outline

表 3. 操作言語のコマンド

Table 3. System Command of Algorithms and Systems

コマンド	説明
INSERT	アルゴリズムを生成する。
SELECT	アルゴリズム・システムを検索する。
CREATE	システムを合成する。
EXECUTE	合成したシステムにデータを代入し、感性検索を実行する。
TEST	システムの合成とデータの代入を同時に行う。(システムの試作支援用)

<式> ::= [<ラベル>, <タイプ>, <値>] |
 [<ラベル>, <タイプ>, <値>, <式>] |
 [<OID>, <タイプ>, <値>] |
 [<OID>, <タイプ>, <値>, <式>] |
 [<値>] | [<値>, <式>]
 <ラベル> ::= 任意の長さの文字列 | % | #
 <%> ::= 識別記号 <#> ::= 識別記号
 <タイプ> ::= 原子型 (integer, string, etc) |
 プログラムの種類 (Java, C, PHP, etc) |
 画像のデータ型 (ppm, bmp, jpg, etc) |
 これらデータタイプの集合 (Set)
 <値> ::= 文字または数字による各種ラベルの値

4.3 アルゴリズムの登録

プラットフォームは、アルゴリズムの登録を図 4 の 1,2 に関係する機構で実現する。開発者は操作言語のコマンド INSERT を用いて、アルゴリズムを登録する。

【システム上の処理の流れ：アルゴリズムの登録】
 (番号は図 4 に対応)

1. アルゴリズムを OEM に変換する : INSERT

```

■ 物理レベルのアルゴリズム M1
(RGB ヒストグラムの生成入力↓)

[postgres@huroRay postgres]$ java ISIS/ISIS/isisCuiMain INSERT M1 SET "[PATH,c,M1]
[INPUT,ppm,Binary:ppm][OUTPUT,file,CommaPhase:double][SIMULATE,String,Physiology]
[KIND,String,ALGORITHM]"

■ 入力情報の解析↓

Input Data=[[PATH, c, M1], [INPUT, ppm, Binary:ppm], [OUTPUT, file, CommaPhase:double],
[SIMULATE, String, Physiology], [KIND, String, ALGORITHM]]

val=/home/postgres/ISIS/ISIS/property ←■ RDB への代入
fileName=/home/postgres/ISIS/ISIS/property
sqlsub=INSERT INTO isictable VALUES(nextval('oid'),'PATH','c','M1')
sqlsub=INSERT INTO isictable VALUES(nextval('oid'),'INPUT','ppm','Binary:ppm')
sqlsub=INSERT INTO isictable VALUES(nextval('oid'),'OUTPUT','file','CommaPhase:double')
sqlsub=INSERT INTO isictable VALUES(nextval('oid'),'SIMULATE','String','Physiology')
sqlsub=INSERT INTO isictable VALUES(nextval('oid'),'KIND','String','ALGORITHM')
INSERT INTO isictable VALUES(nextval('oid'),'M1','SET','[[{1}], [{2}], [{3}], [{4}],
[{5}]]')
objInfo[0]=M1
objInfo[1]=SET
objInfo[2]=[PATH,c,M1][INPUT,ppm,Binary:ppm][OUTPUT,file,CommaPhase:double][SIMULATE,String,Physiology][KIND,String,ALGORITHM]
objInfo[3]=[[{6}]]

←■ アルゴリズム M1 の ID

```

図 5. アルゴリズムの生成例
Fig 5. Inserting Algorithm by INSERT Command

```

■ 認知レベルの感性検索システム A1 の合成入力↓

[postgres@huroRay postgres]$ java ISIS/ISIS/isisCuiMain TEST Feature KIND=Art,User_Name=Ogino
[[1],SET,X][2],file,/data/contents/hrtPPM/hrtPPM.txt[3],text,ISIS/prog/gfe1[4],file,/home/postgre
y/ISIS/tpData/20040619.txt[[14],file,#][21],SET,X][6],file,/home/postgres/f-charts.txt[8],file,#]

val=/home/postgres/ISIS/ISIS/property
fileName=/home/postgres/ISIS/ISIS/property
dbConn=[[1],stackDatafile,SET,[[{6}],[{7}],[{8}],[{9}],[{10}]]]]
sqlObj[[16],PATH,java_stackDatafile],[7],INPUT,file,userinput],[8],INPUT,text,programNameForExecute],[9],INPUT,file,outputfile],[10],OUTPUT,file,outputfile]]
(==SET)SET
##=[1],SET,X]
##=[X],file,/data/contents/hrtPPM/hrtPPM.txt]
##=[X],text,ISIS/prog/gfe1]
##=[X],file,/home/postgres/ISIS/tpData/20040619.txt]
###PersonName=3
tpData=file,text,file]
tpData.size()=3
tpData.size()=3
sqlObj[7],INPUT,file,userinput]
####
(==X)=/data/contents/hrtPPM/hrtPPM.txt
(=)=/data/contents/hrtPPM/hrtPPM.txt
sqlObj[8],INPUT,text,programNameForExecute]
####

←■ 生理レベル(システムは認知・心理・認知で構成)のアルゴリズム M8(3 点間コントラスト特徴)の実行

```

図 6. 認知レベルの感性検索システムの実行例
Fig 6. Executing System by TEST Command

開発者は、C や Java 等で実装された既存・新規のアルゴリズムを、OEM 表現に表現することができる。

2. アルゴリズムのデータベース化

プラットフォームは、OEM 表現されたアルゴリズムを、モデル画像アプローチ[7]に基づいて、関係データベースにマッピングし、蓄積する。これによりプラットフォームは、アルゴリズムの再利用を可能としている。

例えば開発者は、以下のような命令を記述することで、物理レベル(Physics)のアルゴリズム M1(図 1)を生成し、データベースを用いて管理できる。

【INSERT コマンドによるアルゴリズムの生成例】

```

INSERT "M2" "物理レベル"
[PATH, C, ./prog/M1]
[INPUT, file, "画像 PPM" ]
[OUTPUT, String, "カンマ区切りの画像特徴量"]
* 本研究では、記号 : [] で表現される単位をプロパティと呼ぶ。

```

4.4 システムの生成

プラットフォームは、システムの生成を図 4 の 3,4,5 に関係する機構で実現する。開発者は、操作言語のコマンド CREATE を用いて、システムを生成する。

【システム上の処理の流れ:システムの生成】

(番号は図 4 に対応)

3. アルゴリズムの選択:SELECT

開発者は、感性のレベルや対応しているデータ型などを入力することにより、システムに利用したいアルゴリズムを選択できる。

4. システムの生成:CREATE

開発者は、実現したい感性検索に適するアルゴリズムを感性の階層性に基づいて組み合わせることで、システムを生成できる。

5. システムのデータベース化

プラットフォームは、アルゴリズム同様、この機能によりシステムの再利用を可能としている。

例えば開発者は、以下のような命令を記述することで、心理レベルの感性検索システム A1 を生成し、データベースを用いて管理できる。

【CREATE コマンドによるシステムの合成例】

```

CREATE "A1" "心理レベル"
[11(アルゴリズム M1 の OID), SET, %]
[% , file, "コンテンツのリスト"]
[14 アルゴリズム M2 の OID), SET, %]
[% ,String, "サンプルをグループ化"][% , file, "#"]

```

ラベル:%は、プロパティがアルゴリズムの入力情報であることを示す。ラベル:#は、そのプロパティの値として、前の処理の結果を用いることを示す。このラベル:%と#により、システムからアルゴリズムへのデータの受け渡しと、各システムのアルゴリズム間のデータ間で受け渡しを自動的に実行する。

4.5 システムの実行

プラットフォームは、システムの実行を図 4 の 6,7 に関係する機構で実現する。開発者によるシステムの合成・実行は、システムの感性検索の精度を向上させるため試行錯誤的に繰り返される。そのため、開発者はシステムをデータベース登録しないで試行を行いたいという要求を持っている。

また、各個人の感性は時系列的に変化するので、開発者は、各個人のそのときの感性に適したデータを利用して感性検索を実現したいという要求を持っている。

表 4. データベース管理したアルゴリズム

Table 4. Implemented Algorithms on Our Platform

名前	感性のレベル	アルゴリズムの種類
RGB カラーヒストグラム	物理	分析
判別分析	心理・認知	モデル
画像形式変換	---	補完用
VRML 特徴	物理	分析
ユークリッド距離計算	全	検索・規定
主成分分析	心理・認知	モデル
正準相関分析	認知	モデル
3 点間コントラスト	生理	分析

表 5. 生成した感性検索システム

Table 5. Implemented Systems on Our Platform

感性検索システム	合成したアルゴリズム	感性レベル
VRML の不変特徴を用いた例示検索	VRML 特徴 + ユークリッド距離	物理
人の視覚特性(色)に関する例示検索	RGB カラーヒストグラム + ユークリッド距離	物理
人の高次自己相関特性を用いた検索	3 点間コントラスト + ユークリッド距離	生理
視覚的な主観的類似度に基づいた類似検索	3 点間コントラスト + 判別分析 + ユークリッド距離	心理
言語的な解釈によるイメージ語検索	3 点間コントラスト + 正準相関分析 + ユークリッド距離	認知

そのため我々は、データベースに保存しないでシステムを試行する(コマンド:TEST)と CREATE で生成したシステムにデータを代入することで実行する方法(コマンド: EXECUTE)の 2 種類のコマンドを用意した。ただし、どちらのコマンドを使っても、処理の仕方、結果は同じである。

【システム上の処理の流れ：システムの実行】

(番号は図 4 に対応)

EXECUTE の例(TEST の場合は 6 を行わず、システムを合成し 7 を試行する)

6. システム選択：SELECT

プラットフォームは、利用者の感性検索の要求に関する入力情報をもとに、感性のレベルやコンテンツが適するシステムを選択する。

7. システムの実行(試行)：EXECUTE(TEST)

プラットフォームは、選択したシステムに、利用者の教示データやコンテンツ等をそれぞれ代入し、その利用者の感性の基準で、感性検索を実行する。

例えば開発者は、以下のような命令を記述することで、感性検索システム A1 の試行や、そのときの感性

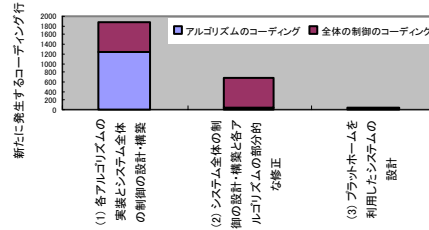


図 7. 新たに発生するコーディング量の比較

Fig 7. Comparing of Program Steps

をあらわすデータを代入しての実行を行える。

【TEST コマンドを利用した A1 の実行例】

```
TEST“A1”“User=’Person1’,Kind=’Art’,...”
[11, SET, %][%, file, “./Content/Art.txt”]
[14, SET, %
[%, string, “1, 1, 4, 5...”][%, file, “#”]
```

【EXECUTE コマンドを利用した A1 の実行】

```
EXECUTE“A1”“User=’Person1’,Kind=’Art’,...”
[“./Content/Art.txt”][“1, 1, 4, 5...”][#]
```

4.6 プラットホームの有効性の評価

プラットフォームの評価方法として、感性検索システムを実装する時のコーディング量を比較した。

- (1) アルゴリズムとシステムを始めから設計・構築する場合
- (2) 既存のアルゴリズムを利用してシステムを設計・構築する場合
- (3) 我々のプラットフォームを利用して設計・構築する場合

(1)は、各レベルのアルゴリズムとシステム全体を制御する部分のコーディング(図 7 では、1000 行以上)が発生するため、システム開発にかなりの労力を必要としている。(2)の方法でも、アルゴリズムの組み合わせるための部分的な修正やシステム全体を制御する部分をコーディング(図 7 では、300 行程度)する必要が出る。

これらの方法に対して我々の(3)は、10 行程度の操作言語の設計により、感性システムを設計できるため、システム開発の労力を大幅に軽減できている。

5. まとめ

本論文では、(1) 感性情報処理アルゴリズムと感性検索システムのデータベースを用いた管理の仕組みと、

(2) このアルゴリズムを組み合わせることによりシステムを動的に生成するための仕組みを提案した。そして我々は、(3) 上記の 2 の機能とその操作言語を持ったソフトウェアプラットフォームを構築した。

(1)に関して我々は、アルゴリズムとシステムを半構造データモデル統一的に記述する方法を定義し、それらをデータベースにより管理する機構を開発することにより実現した。

(2)に関して我々は、感性情報処理アルゴリズムを組み合わせることで感性的検索システムを動的に合成する機構とその操作言語を設計・開発することで実現した。

(3)により我々は、個人、コンテンツ、検索の仕方に対応させて、視覚的な類似性や言語的な解釈による感性的などによる検索に対応した感性的検索システムの動的な生成することを可能とした。

本プラットフォームは、オープンソースとして公開するため、その文書化を現在行っている。

参 考 文 献

- 1) 西尾, 田中他, “情報の構造化と検索,” 岩波書店, pp.168-221, 2000年3月
- 2) 清木, 金子, 北川, “意味の数学モデルによる画像データベース探索方式とその学習機構,” 電子情報通信学会論文誌, Vol.J79-D-2, No.4, pp.509-519, April(1996).
- 3) 木本, “感性的語による画像検索とその精度評価,” 情報処理学会論文誌, Vol.40, No.3, pp.886-898, (1999).
- 4) 加藤, 栗田, “画像の内容検索, 電子美術館への応用,” 情報処理, Vol.33, No.5, pp.466-477, May(1992).
- 5) 田島, “半構造データのためのデータモデルと操作言語,” 情報処理学会論文誌:データベース, Vol. 40, No. SIG3(TOD1), pp. 152-170, 1999
- 6) D. Quass, A. Rajaraman, Y. Sagiv, J. Ullman, and J. Widom, "Querying Semistructured Heterogeneous Information", In International Conference on Deductive and Object-Oriented Databases, 1995.
- 7) 吉川, “データベースの観点から見た XML の研究,” 2002年情報学シンポジウム講演論文集, pp.25-31,2002
- 8) 油井, 森嶋, “PostgreSQLを用いた多機能なXMLデータベース環境の構築,” 情報処理学会論文誌:データベース, Vol.40, No.SIG12(TOD19), pp.11-22, Sep.2003
- 9) 荻野, 加藤, “個人の主観的な解釈に適合した情報サービスを提供するためのソフトウェアプラットフォームの構築,” 2004 映情学技報 ME/AIT, pp.45-48, May.2004.