

# Software Platform for Information Service Systems Suitable for Individual Subjective Interpretation Process \*

**Akihiro OGINO**

Graduate School of Science and Engineering  
CHUO University  
Tokyo, Japan  
ogino@indsys.chuo-u.ac.jp

**Toshikazu KATO**

Faculty of Science and Engineering  
CHUO University  
Tokyo, Japan  
kato@indsys.chuo-u.ac.jp

**Abstract** - We proposed a software platform for developing subjective similarity image retrieval systems that provide information to individual users suitable for our subjective interpretation process to contents, such as art, scenery, etc. Our platform has three mechanisms: (1) A mechanism for performing a system synthesis by combining modelling algorithms of subjective interpretation process (2) A mechanism for managing created application systems, algorithms and data that are contents, individual profile and individual's interpretation teaching data to images, by semi-structured data model and database (3) A mechanism for operating the platform through commands for developing and managing application systems in a uniform manner. We applied many image formats and a VRML format to the platform. We inserted algorithms of statistical analysis and content's features calculation into the platform. We developed different types of application system on platform. We have only to choose and combine algorithms on the platform to develop application systems with less effort.

**Keywords:** Cognitive systems engineering, application of human senses and perception to engineering: KANSEI engineering, software platform, semi-structured data model.

## 1 Introduction

Recently, subjective similarity image retrieval systems are becoming popular, which are for arts, photos, clothes, and pieces of furniture. We describe subjective similarity image retrieval systems as application systems in this paper.

To service images suitable for my own subjective interpretation for the individual, we have to realize three things:

1. Modelling of subjective interpretation processes to each user
2. Using many kinds of images
3. Servicing to each user by many retrieval methods

However, existing application systems only model a subjective interpretation process of a specific individual or a specific image category [1]. Current application systems only service a specific retrieval method that is designed for their images and for a single similarity criterion. Therefore, these application systems can't correspond with the interpretations of each user or many image categories and many retrieval methods.

In this paper, we try to solve these problems by two methods:

1. Method for managing synthesized application systems, algorithms and data, which are images, individual profile, individual's interpretation teaching data to images and models of subjective interpretation processes, by semi-structured data models and databases
2. Method for performing an application system synthesis by combining modelling algorithms of subjective interpretation processes

We developed a software platform with functions to manage algorithms and data, and to create application systems. We also designed commands for operating the platform.

## 2 Subjective Interpretation Process

### 2.1 Explanation of Subjective Interpretation Process

We have proposed that subjective interpretation processes have four levels [1].

#### 1. Physical Level

A visual cue may often remind us of similar contents. This process is a kind of similarity and associative retrieval of contents by physical level interaction with the contents database.

We can retrieve contents that have the same composition to a retrieval key from an images database

Table 1. Types of Modelling Algorithms of Subjective Interpretation Process

Types of Algorithms	Explanation
Analysis Algorithm	To analyse and quantify data, such as image's features or individual's interpretation teaching data to images
Model Algorithm	To classify data, which was analysed and quantified by analysis algorithm, based on individual similarity criterion and to map the classified data to a high level of subjective interpretation process
Retrieval Algorithm	To retrieve the non-classified data or the classified data

Table 2. Attribute Label of OEM for Algorithm, System

Label	Explanation
SIMULATE	Level of subjective interpretation process
INPUT	Data type and Argument of algorithm
OUTPUT	Data type and Argument of algorithm
PATH	File path of algorithm's program
KIND	Information which the OEM object shows is an algorithm or a system

Table 3. Attribute Label of OEM for Data

Label	Explanation
USER	User's profile, such as Name, Sex, Age, etc
KIND	Information which the OEM object shows is data
CATEGORY	Contents categories, such as Art, Furniture, Symbols, etc
MODEL	Model of subjective interpretation process about individual user
SITUATION	A time and place which performed a supervised learning activity or created the model
FEATURE	Contents of graphical feature
BASE	Information on the system which the model created

using the composition of an image that the user drew as a retrieval key, by modelling of this level.

## 2. Physiological Level

Early stages of mammal vision mechanisms extract graphical features of contents, such as intensity levels, edge, contrast, correlation, spatial frequency, and so on. Visual perception may depend on such parameterised graphical features.

We can retrieve contents that are similar to the retrieval image key from a contents database using an image sketch that the user drew as a retrieval key, by modelling at this level.

## 3. Psychological Level

We have to notice that the criteria for similarity belong to a subjective human factor. Although human beings have anatomically the same structure, each person may show different interpretations in classification and similarity evaluation. It means each person has their own weighting rules on graphical features. We describe converted graphical features by our own rules as subjective features.

We can retrieve contents that are similar to retrieval image key based on my own similarity interpretation from contents database using an image sketch that the user drew or a sample image as a retrieval key, by modelling at this level.

## 4. Cognitive Level

We often have different impressions, even when viewing the same painting. Each person may also give a unique interpretation even when viewing the same picture. It seems each person has their own correlation rules between concepts and graphical features and/or subjective features.

We can retrieve contents that are similar to a retrieval description key based on our own impression interpretation

from the contents database using individual's descriptions about impression as a retrieval key.

We describe data of a modelled individual subjective interpretation processes as a subjective interpretation model. We describe each user's profile and each user's interpretation teaching data to images and subjective interpretation models as KANSEI data in this paper.

## 2.2 Necessary of Software platform

System developers are designing and coding many kinds of algorithms of each level. System developers are building these subjective similarity retrieval systems using different algorithm level. System developers are using different image category and interpretation teaching data to images in discrete application systems, respectively.

Therefore, if there is a software platform that can perform an integrated management of algorithms, application systems, images and data for modelling interpretation process and can create application systems by combining many kinds of algorithms, system developers can create many types of subjective similarity image retrieval systems and try modelling experiments using many kinds of data.

We need a software platform that has the following functions:

1. A function that can integrate and manage application systems, algorithms, images and KANSEI data
2. A function that can combine algorithms at each level and develop it as application systems

### 3 Management Method

There are following problems about integrated management of many kinds of algorithms, application systems, images and KANSEI data.

1. Each retrieval method uses different kinds and types of images and KANSEI data
2. Each modelling algorithm of the subjective interpretation uses a different kind, type and number of images and KANSEI data
3. Each retrieval system, which have different retrieval methods

We have to express algorithms, application systems, images, KANSEI data, using a common data model. We also have to manage them by database.

We defined a common data model for the managing of images, KANSEI data, algorithms and application systems, using semi-structured data models.

#### 3.1 Definition of Common Data model based on Semi-Structured Data Model

The semi-structured data model is a self-descriptive and nested structure. This model can manage different type of structured data, such as user profiles that have different kinds and numbers of attributes by individual or application.

We used OEM (Object Exchange Model)[2],[3], one of the semi-structured data models in this study. OEM is a pure database model and the most popular for semi-structured data. We added a new data type to OEM for the management of images, KANSEI data, algorithms and application systems, and used it as a common data model.

[Definition 1: Common Data model based on OEM]

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

- *Object-ID*: A unique variable-length identifier for object (not null).
- *Label*: A variable-length character string describing what the object represents.
- *Type*: The data type of the object's value. Each type is any one of an atom type (such as integer, string, etc), images type (such as jpg, ppm, etc), real-world data type (such as time, date, etc), program type (such as Java, C, etc), or the type set
- *Value*: A variable-length value for the object.

We defined attribute labels of OEM for management images, KANSEI data, algorithms and application systems when modelling subjective interpretation processes, in Table 2, 3.

```
/* Information of Individual User */
Subjective Interpretation Model U1 :: =
< Id13, U1, set, {Id13-1, Id13-2, Id13-3, Id13-4,
                  Id13-5, Id13-6, Id13-7}>

/* Kind of OEM object */
< Id13-1, KIND, string, "Model">

/* User's Profile */
< Id13-2, USER, set, {Id13-2-1, Id13-2-2}>
< Id13-2-1, NAME, string, "Ogino Akihiro">
< Id13-2-2, Age, string, "28">

/* The time and place which performed
a supervised learning or created the model */
< Id13-3, SITUATION, set, {Id13-3-1, Id13-3-2}>
...
< Id13-4, SIMULATE, string, "Psychological">
< Id13-5, BASE, set, {Id13-5-1, Id13-5-2, Id13-5-3}>
...
```

Figure 1. Example of KANSEI data by OEM

```
/* Algorithm of calculating color histogram of image */
Analysis Algorithm M1 :: =
< Id1, M1, set, {Id1-1, Id1-2, Id1-3, Id1-4, Id1-5}>

/* Kind of OEM object */
< Id1-1, KIND, string, "Algorithm">

/* The level of subjective interpretation process
which an algorithm simulates */
< Id1-2, SIMULATE, string, "Physical">

/* The path of algorithm's program */
< Id1-3, PATH, C, /algorithm/M1>

/* Input of the Algorithm's program */
< Id1-4, INPUT, ppm, "a image of ppm data type">

/* Output of the Algorithm's program */
< Id1-5, OUTPUT, file, "comma-delimited file">
```

Figure 2. Example of Algorithm by OEM

```
/* Subjective Similarity Retrieval System */
System A1 :: =
< Id5, A1, set, {Id5-1, Id5-2, Id5-3, Id5-4, Id5-5}>

/* Kind of OEM object */
< Id5-1, KIND, string, "System">

/* The level of subjective interpretation process
which an algorithm simulates */
< Id5-2, SIMULATE, string, "Psychological">

/* First processing (color histogram calculating of image) */
< Id5-3, M1, set, {Id5-3-1, Id5-3-2}>
< Id5-3-1, PATH, C, /algorithm/M1>
< Id5-3-2, INPUT, file, "Image(type:ppm)">

/* Second processing (discriminant analysis) */
< Id5-4, M6, set, {Id5-4-1, Id5-4-2, Id5-4-3}>
< Id5-4-1, PATH, C, /algorithm/M6>
< Id5-4-2, INPUT, file, "Image(type:ppm)">
< Id5-4-3, INPUT, file, "Supervised learning result
(comma-delimited file)">

/* Third processing (Euclidean norm calculating) */
< Id5-5, M11, set, {Id5-5-1, Id5-5-2}>
< Id5-5-1, PATH, C, /algorithm/M11>
< Id5-5-2, INPUT, file, "Subjective features
(comma-delimited file)">
```

Figure 3. Example of System by OEM

id	label	type	value
1	PATH	c	M1
2	INPUT	ppm	Binary:ppm
3	OUTPUT	file	CommaPhase:double
4	SIMULATE	String	Physiology
5	KIND	String	ALGORITHM
6	M1	SET	[[[1]], [[2]], [[3]], [[4]], [[5]]]
7	PATH	c	M2
8	INPUT	file	CommaPhase:double
9	INPUT	file	CommaPhase:int
10	OUTPUT	file	CommaPhase:double
11	SIMULATE	String	Physiology
12	KIND	String	ALGORITHM
13	M2	SET	[[[7]], [[8]], [[9]], [[10]], [[11]], [[12]]]
14	PATH	c	M3
15	INPUT	file	CommaPhase:double
16	INPUT	file	outputfile
17	OUTPUT	file	outputfile
18	M3:PCA	SET	[[[14]], [[15]], [[16]], [[17]]]
19	PATH	c	M4
20	INPUT	file	1
21	INPUT	file	1

Figure 4. Example of Mapping Algorithm to RDB

### 3.2 Notation by Common Data Model

In this chapter, we describe examples of the OEM object about image, KANSEI data, algorithm, and application system.

Figure 1 shows an example of the subjective interpretation model of individual users. We create OEM objects of KANSEI data for individual users. We create and accumulate new OEM objects of KANSEI data for each user by changing individual's subjective interpretation.

We can analyse a direction or average of an individual's subjective interpretation using stored data. We can also create OEM objects of images by image's features, respectively.

Figure 2 shows an example of a modelling algorithm for calculating graphical features about an image's colour histogram. Figure 3 shows an example of an application system at the psychological level. We create each OEM objects of algorithm and application system by kind, respectively. We accumulate them and can have many algorithms and application systems on a platform. We can service a retrieval of images suitable for subjective interpretation to the individual using many kinds of retrieval methods.

By the OEM object for subjective interpretation process modelling, system developers have only to put different kinds of data, algorithm and system into a common data model to manage them.

### 3.3 Mapping to Relational Database

In this study, we manage data, algorithms and application systems by mapping OEM objects to the relational database (RDB). We use a model mapping approach [4] to map OEM object to RDB.

Figure 4 shows an example of mapped OEM algorithms in a RDB. We map OEM objects in a relational table that has an oid (Object-ID), a label, a type and a value as attribute of its table.

We could add and update of data, algorithms and application systems by managing them using this database.

## 4 Software Platform

We developed a software platform for building subjective similarity retrieval application systems. Our platform has three mechanisms:

1. A mechanism for performing a system synthesis by combining modelling algorithms of subjective interpretation process
2. A mechanism for managing created application systems, algorithms and data, which consist of contents, individual profile and individual's interpretation teaching data to images, by semi-structured data models and databases

3. A mechanism of operating the platform by commands for developing and managing application systems in a simple manner

We developed these mechanisms using Java and PostgreSQL on Linux. We showed an outline of our platform in Figure 6. We use the platform through a console of Linux.

We inserted ten kinds of images, into our platform (Table 5). We inserted five algorithms (Table 6). We used KANSEI data of five students, three men and two women.

### 4.1 Operation Commands on Software Platform

We defined commands for the software platform management. Users can create application systems by combining modelling algorithms and manage many kinds and types of algorithms, application systems, images and KANSEI data using these commands in a simple manner. We showed the commands and their functions in table 4.

[Definition 2: Operation command]

Operation Command ::=

<Command><Label><Additional information>  
"<Expression>"

- Command ::= CREATE, INSERT, EXECUTE, TEST
- Label ::= A variable-length character | Mark
- Additional information ::=  
    <Label = Value> |  
    <Label = Value>, <Additional information>
- Expression ::=  
    "[<Label>, <Type>, <Value>]" |  
    "[<Label>, <Type>, <Expression>]" |  
    "[<Label>, <Type>, <Value>]", "<Expression>" |  
    "[<OID>, <Type>, <Value>]" |  
    "[<OID>, <Type>, <Value>]", "<Expression>" |  
    "[<Value>]" | "[<Value>]", <Expression>
- Type ::=  
    Each type is any one of an atom type |  
    Images type | VRML type |  
    Real-world data type | Program type | the type set
- Mark ::= # | %
- Value ::= A variable-length value for the object | Mark | Expression

### 4.2 Insert Data and Algorithms

#### 4.2.1 INSERT Process on Platform

Our platform inserts algorithms, images and KANSEI data by the inserting function that we showed in Figure 5. Following numbers correspond to numbers in Figure 5.

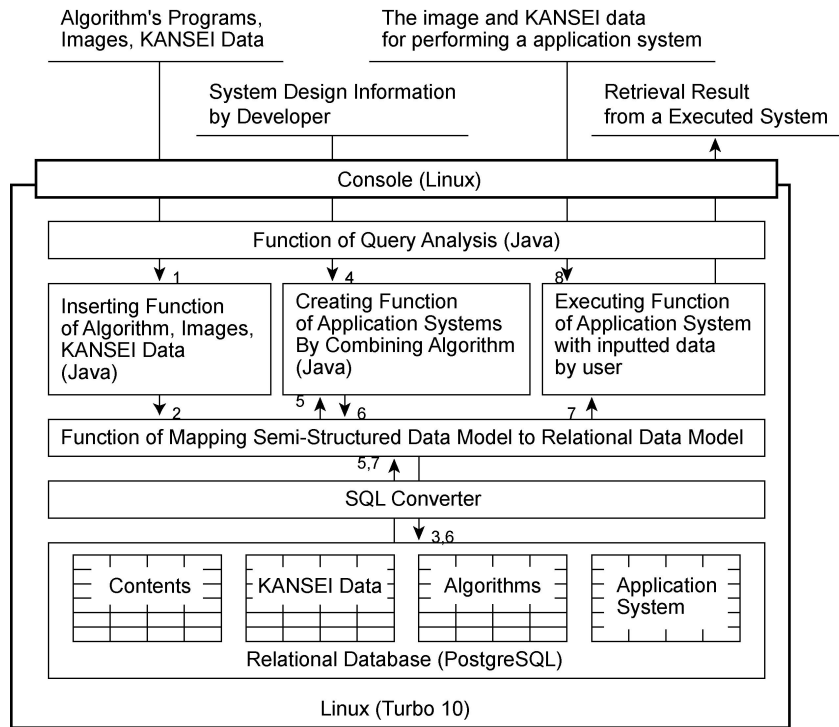


Figure 5. Outline of our software platform

Table 4. Operation Command of our platform

Operating Command	Explanation
INSERT	To insert algorithms, images and KANSEI data into a platform
CREATE	To create application systems by combing algorithms on platform
SELECT	To select algorithms, systems, images and KANSEI data from the platform
EXECUTE	To execute application system by setting image and KANSEI data suitable for individual subjective interpretation processes
TEST	To create and execute application system with sample data

#### [INSERT Process on Platform]

1. Input Algorithms, Images and KANSEI data  
User can insert algorithms, images and KANSEI data into the platform through the same interface. The user can also insert new or existing algorithm programs that are coded by C, Java, etc, and many kinds and types of images that are jpg, bmp, ppm, into the platform.
2. Convert to OEM objects  
Our platform converts all kinds of algorithms, images and KANSEI data to OEM objects with this mapping function. This function expresses every kind of algorithm, data and image as an OEM object, respectively.
3. Map to RDB  
The platform maps OEM objects to RDB (PostgreSQL) based on the model mapping approach. The platform accumulates many kinds of OEM objects. The platform can reuse of algorithms, images and KANSEI data through management of OEM objects with a database.

#### 4.2.2 Example of INSERT Process

The user can insert algorithms, images and KANSEI data into the platform with an INSERT command.

For example, a user can insert an analysis algorithm at a physics level was shown in Figure 2, into the platform. The formula of the INSERT command is described as follows.

[Example1: INSERT Command]  
 INSERT "Subjective interpretation model U1"  
 "SIMULATE=physical, KIND=model"  
 [USER, set,  
 {[Name, string, "Ogino"], [Sex, string, "Men"]}]  
 [CATEGORY, string, "Art"]  
 [MODEL, set,  
 {[Parameter1, string, 0.0532, 1.6234...],  
 [Parameter2, string, 0.72457, 3.620...]}...]  
 [SITUATION, set, ]

Table 5. Implemented Algorithms on Our Platform

Algorithm Name	Level of Interpretation Process	Algorithm Kind
RGB Colour Histogram	Physical	Analysis
Discrimination analysis	Psychological , Cognitive	Model
VRML Feature Calculate	Physical	Analysis
Euclidean distance Calculate	All	Retrieval
Principal Component Analysis	Psychological , Cognitive	Model
Three-point Contrast Feature Calculate	Physiological	Analysis
Canonical Analysis	Cognitive	Model
Image Type Convert	---	Subservience

[Example2: INSERT Command]

INSERT "Algorithm M2"

"SIMULATE=physical, KIND=algorithm"

[PATH, c, /algorithm/M1]

[INPUT, file, "ppm"]

[OUTPUT, file, "Subjective interpretation model

(Comma-delimited)"]

We describe a unit expressed with "[ ]" of OEM object as a property.

## 4.3 Create Systems

### 4.3.1 CREATE Process on Platform

Our platform creates application systems by the creating function that we illustrated in Figure 5. Following numbers correspond to numbers in Figure 5.

[CREATE Process on Platform]

#### 4. Design of System

A user can design a processing flow of an application system by combining algorithms, based on the subjective interpretation process. A user can design many kinds of application system suitable for their own system design in a uniform manner.

#### 5. Select and Combine Algorithms and Create systems

The creating function automatically selects algorithm from a database and combines them based on the system design of its users. The creating function creates an application system as OEM objects.

Table 6. Implemented Application Systems on Our Platform

Application System	Combined Algorithms	Subjective Interpretation Process Level
Sketch retrieval with eternal feature about VRML	VRML features +Euclidean distance	Physical
Sketch retrieval with RGB colour feature of images	RGB colour histogram features +Euclidean distance	Physical
Sketch retrieval with autocorrelation of a high order	Three-point contrast features +Euclidean distance	Physiological
Similarity retrieval based on a visual subjective similarity	Three-point contrast features +Discrimination analysis +Euclidean distance	Psychological
Image-word retrieval with linguistic interpretation	Three-point contrast features +Canonical Analysis +Euclidean distance	Cognitive

#### 6. Map to RDB

Users can use many kinds of application systems through management of OEM objects with the database.

### 4.3.2 Example of CREATE Process

The user can create application systems on the platform with the CREATE command.

For example, a user can create physiological application system, which was shown in Figure 3, on the platform. The formula of the CREATE command is described as follows by a user.

[Example3: CREATE Command]

CREATE "System A1"

"SIMULATE= Psychological, KIND=system"

[11(The oid of algorithm M1), set, %]\*\*

[%, file, "A list of path about ppm images"]\*\*\*

[14(The oid of algorithm M6), set, %]

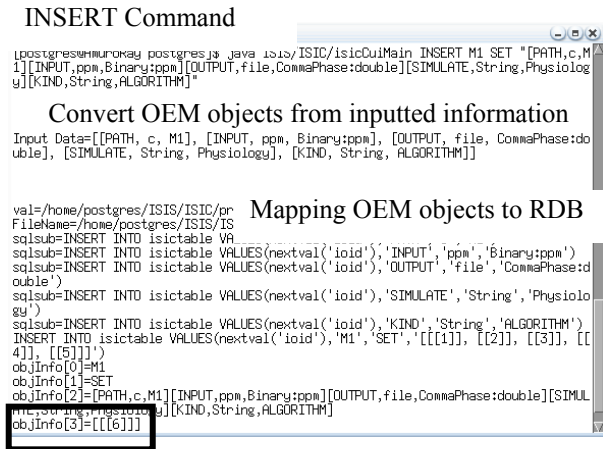
[%, file, "a supervised leaning data

(Comma-delimited)"]

[%, file, "#"]

When a property has a % value, such as \*\* in Example 3, the property has some properties as set of values. When a property has a % label, such as \*\*\* in Example 3, the property shows input parameter. When a property has a # value, the property shows a result of a previous processing value.

Our platform can automatically perform a passing of data or parameters from a system to algorithm's programs and from one algorithm to another by the % and # marks.



Oid of Algorithm M1 (No.6)

Figure.6 Example of Inserting Algorithm into Our Platform by INSERT Command

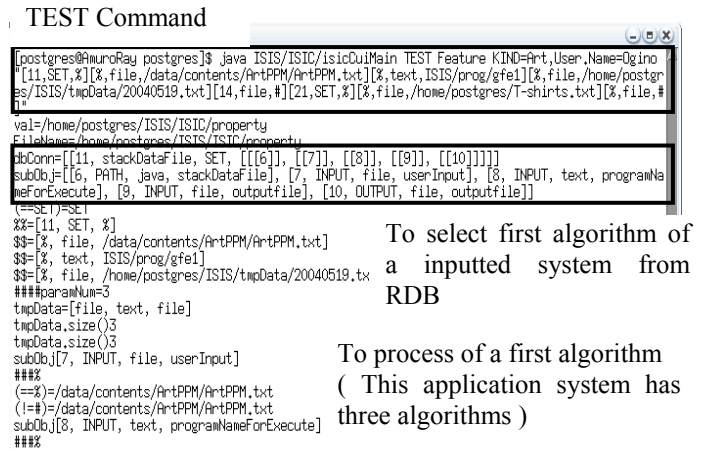


Figure.7 Example of Performing System on Our Platform by TEST Command

## 4.4 Execute Systems

Our platform executes application system with inputted images and KANSEI data by the executing function that we showed in Figure 5. A system developer repeats the process of trial and error in developments of application systems as a means of improving the precision of subjective interpretation process modelling or subjective similarity image retrieval. Therefore, our platform has two commands, TEST and CREATE.

The TEST command test systems without accumulating them. Therefore, system developers can perform a trial of application systems with sample images and KANSEI data.

The CREATE command executes systems with inputted images and KANSEI data. System developers can change images and KANSEI data, which are for executing systems, by modelling the subjective interpretation processes according to their purpose. Therefore, system developers can execute systems with new inputted images, new KANSEI data, which varied with time, other image categories and other individual's KANSEI data.

### 4.4.1 Execute Process on Platform

Our platform creates application system by a creating function that we showed in Figure 5. Following numbers correspond to numbers in Figure 5.

[EXECUTE Process on Platform]

#### 7. Select System

Users can select an application system for performing an image retrieval based on individual subjective

interpretation processes from the database that has application systems of all levels.

#### 8. Input Images and KANEI data into System

Users can perform subjective similarity image retrieval by inputting images and KANSEI data into a selected system. Therefore, users can perform different subjective similarity image retrieval on the same application system by changing images and KANSEI data.

User can execute application systems on the platform with an EXECUTE or TEST command.

### 4.4.2 Example of EXECUTE Process

For example, a user can execute or test physiological application system, which was shown in Figure 3, on the platform. The formula of the EXECUTE or TEST command is described as follows by a user. TEST command does two things, it performs a CREATE and EXECUTE at once.

[Example4: TEST Command]

TEST "System A1"

"USER='Ogino', CATEGORY='Art' ..."

[11, set, %][%,file, "/Images/ArtList.txt"]

[14, set, %]

[%,file, "/Leaning/OginotoArt.txt"] [%,file, "#"]

[Example5: EXECUTE Command]

EXECUTE "Oid:113(System A1's Oid)"

"USER='Ogino', CATEGORY='Art' ..."

["/Images/ArtList.txt"]

["/Leaning/OginotoArt.txt"] ["#"]



## 4.5 Evaluation of Our Platform

We compared the three methods in terms of the number of coding steps, which are needed to develop an application system.

1. Method of developing algorithms and systems from zero
2. Method of developing systems using existing algorithms
3. Method of developing systems using our platform

By development of an application system based on method (1), we were needed to write 5,000 or more lines. Especially, it was difficult for us to design and develop an algorithm and a system together. Therefore, we needed more effort for the developing application systems.

By development of an application system based on method (2), we needed to write 300 or more lines to change algorithms and develop all of the system. It was difficult for us to change other users' developed algorithms, in order to incorporate them into the application systems.

By development of an application system using our platform that is method (3), we could develop an application system by writing the command of only 10 lines.

We could develop the application systems of Table 5 in just a matter of minutes, by inputting and combining algorithms, and writing and executing an operation command, using our platform. Our platform could develop application systems with less effort than other methods. Especially, it is a good point of our platform that we do not need to build a system for every level of an algorithm or a subjective interpretation process.

## 5 Conclusions

In this paper, we proposed a software platform for developing subjective similarity image retrieval systems that provide information to individual users suitable for their subjective interpretation process. Our platform had three mechanisms:

- (1) A mechanism for managing created application systems, algorithms and data that are contents, individual profiles and individuals' interpretation teaching data to images, by semi-structured data models and databases
- (2) A mechanism for performing a system synthesis by combining modelling algorithms of subjective interpretation process.
- (3) A mechanism for operating the platform with commands for developing and managing application systems in a uniform manner.

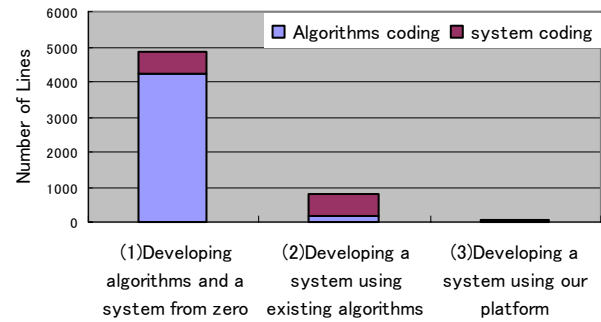


Figure 8. Comparison of program lines

We developed mechanism (1) by describing algorithms, systems, images and KANSEI data using semi-structured data models and managing them using relational database (PostgreSQL). We realized the platform, which has many kinds of systems, algorithms, images and KANSEI data. We could service images suitable for individual subjective interpretation processes to each user by utilising many kinds of retrieval methods.

We developed mechanism (2) by building a function that combines the algorithms of OEM objects by user's system design based on the subjective interpretation processes. We could develop application system in a simple manner, because we developed the function that performs automatically a passing of data or parameters from a system to algorithm's programs and from one algorithm to another algorithm.

We developed mechanism (3), which has five operating commands for developing and managing application systems, using Java. We could develop application systems, which have different retrieval methods, images and KANSEI data respectively, on our platform with less effort and a few minutes.

We are documenting our platform's usage and we will release the content of our platform as an open source.

## References

- [1] Kyoji Hirata, Toshikazu Kato, "Query by Visual Example - Content based Image Retrieval", the 3rd International Conference on Extending Database Technology: Advances in Database Technology, 56 - 71, 1992
- [2] 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.
- [3] <http://www-db.stanford.edu/tsimmis/>
- [4] Masatoshi Yoshikawa, Toshiyuki Amagasa, Takeyuki, Shimura and Shunsuke Uemura. "{XRel}: a path-based approach to storage and retrieval of {XML} documents using relational databases". Transactions on Internet Technology, Vol 1- Num 1, pp.110-141, August, 2001