Transactions of the Materials Research Society of Japan 29 [1] 293-296 (2004)

Multi-dimensional data management by virtual sample library written in object-oriented script language *Ruby*

Shin-ichi Todoroki* and Satoru Inoue
Advanced Materials Laboratory, National Institute for Material Science
1-1 Namiki, Tsukuba, Ibaraki 305-0044, Japan
Fax: 81-29-854-9060, e-mail: TODOROKI.Shin-ichi@nims.go.jp

Virtual sample library (VSL) is developed for accessing multi-dimensional data acquired from several measurements on a single combinatorial sample library. Management of such data is generally very hard because each data is stored independently in various format. The data is structuralized and standardized by VSL, which has a hierarchy structure whose top layer has the same geometry of the combinatorial sample library. Each data is stored into the VSL systematically according to their coordinates in the library, the name of the measurements, and the dimension of the data. Thus, VSL can provide any data for analysis and visualization by specifying their identifying information.

An actual example is demonstrated for the case of one-dimensional combinatorial glass sample library. Some tellurite glass libraries containing Er and F are annealed in a temperature-gradient furnace and the decay curves of 1.5μm-fluorescence of Er^{3+} are recorded along the libraries in order to find annealing conditions for precipitation of Er-containing fluoride crystals. Their fluorescence spectra, fluorescence lifetime, state of precipitates as a function of annealing conditions (7-dimensional data) are plotted in two figures through VSL.

Key words: multi-dimensional data, virtual sample library, object-oriented, informatics

1. INTRODUCTION

Combinatorial technology brings about significant advantages to our research activities provided we can afford to analyze large amount of data obtained. When we are to make several kinds of measurements on one combinatorially-integrated sample library, the obtained data become multi-dimensional and quite hard to deal with by ordinary spreadsheet programs. For example, suppose that we have a 2-dimensional(2D) sample array on which three kinds of measurements are performed, and each measurement gives us data in scalar, 2D, and 3D format per pixel, respectively. This situation is illustrated in Fig. 1. Then we obtain 3D, 4D and 5D data from these measurements.

Since the coordinates in the sample array are temporary parameters for us, they should be converted to appropriate physical quantities (e.g. composition, annealing temperature, etc...) according to the fabrication condition of the sample library. Moreover, when we need to see the correlation between these measurements, we have to merge and re-compile the whole data so that the individual data measured at the same pixel are related.

These time-consuming editing jobs would be reduced if we could treat the whole data in one format, in which each data is linked with the corresponding coordinate and physical quantity. This paper demonstrates that such a multi-dimensional data management is possible through "virtual sample library"(VSL), which is a data medium used in software programs for analysis and/or visualization.

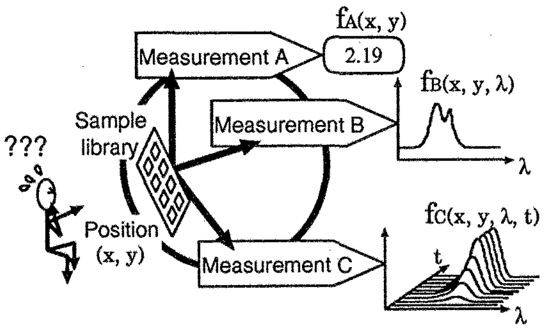

Fig. 1: Illustration showing a situation of treating multi-dimensional data obtained from several measurements on a combinatorially-integrated sample library (see text).

2. WORKING HYPOTHESIS

First of all, let us discuss what kind of format is feasible for us to store the whole multi-dimensional data and access any of them easily. Considering the coordinates in the library are common information among the data obtained from each measurement, it is reasonable to store the data under the individual coordinates hierarchically. After specifying an arbitrary position in the library, we naturally notice what kind of measurements are performed there or what is the fabrication condition there. Finally, we can access data for the specified measurement or fabrication condition.

One example of this hierarchy structure is visualized

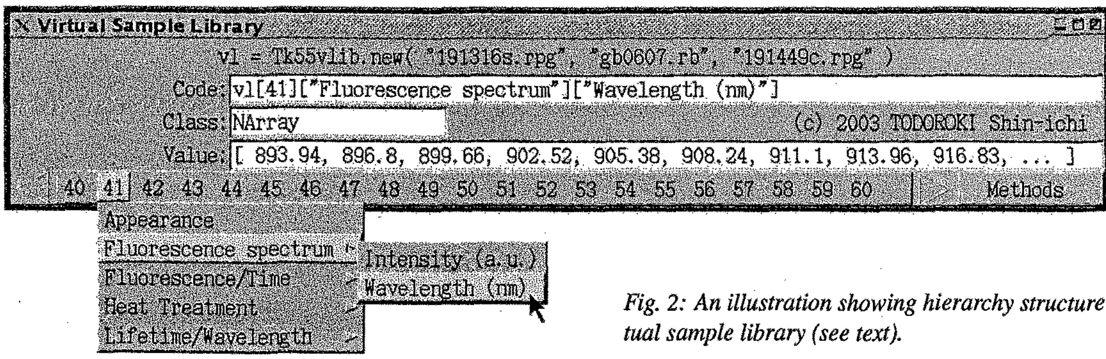

Fig. 2: An illustration showing hierarchy structure of virtual sample library (see text).

as "pull-down menu" style in Fig. 2, where an 1D sample array is used as a model. In the bottom of the window, a menu is located showing the coordinates in the library, from 40 to 60. After the item of "41" is selected, a sub-menu appears showing what kinds of data are stored. Next, the item of "Fluorescence Spectrum" is chosen and another sub-menu is opened to show this consists of two items. Finally, by choosing "Wavelength" item, a series of wavelength values are shown at "Value" box just above the main menu.

In this way, such a hierarchy structure can store the whole multi-dimensional data systematically and gives us an easy way to access individual data. Let us call this structure, "virtual sample library" (VSL), since the top of the hierarchy corresponds the shape of the sample library.

The next thing we have to consider is how we construct this VSL and how we use it for data analysis and visualization. Although there are many ways to realize it, we decided to make it as an original software written by object-oriented script language, called *Ruby*[1, 2]. This is because object-oriented languages have abstract expressive power essential for treating complicated data[3] and *Ruby* was developed for reducing the work of programmers and allowing them concentrated on substantial matters[1].

3. METHODS AND PROCEDURES

For constructing VSL, a set of data is used which was collected from 1D sample arrays of Er^{3+}-F-co-doped tellurite glass[4]. The samples of about 30 cm in length were prepared by sucking the glass melt into a pre-heated glass capillary in a vertical temperature-gradient furnace and were annealed in the furnace and/or quenched with different conditions; (0) no annealing, (1) being annealed for 5min, and (2) another successive annealing at a different position in the furnace for 5min. Since the annealing temperature is a function of the height in the furnace, 1st and 2nd annealing temperatures for the sample (1) and (2) are calculated according to the temperature profile in the furnace. Appearance of these glass samples are judged by human eye, as "transparent", "white" or "opaque". Time-resolved fluorescence intensity of Er^{3+} ions (1.5μm band, excitation: 977nm) were collected along the sample libraries in every 1mm.

The collected data of about 400MB in size are converted to obtain positional dependence of heat treatment temperatures, appearance, fluorescence spectra (CW component), and lifetime of 1.533μm fluorescence. Sum of their dimension is 7, which is listed in Table 1, and the total size of data file is about 2.7MB in text format.

Table 1: List of the 7-dimensional data used in this study (see text).. Some of these are plotted in Fig. 3 and 4.

#	Data	Fig. 3	Fig. 4
1	Position, x/mm	✓	✓
2	1st annealing temp., T_1/°C	✓	✓
3	2nd annealing temp., T_2/°C	✓	
4	Appearance of glass segment	✓	
5	Fluorescence lifetime, τ/msec		✓
	Fluorescence spectra (CW component)		
6	Intensity (a.u.)		✓
7	Wavelength, λ/nm		✓

The VSL is composed of three parts; (a) the definition of the top of the hierarchy, (b) the procedure of loading experimental data and storing them under the top of the hierarchy, and (c) the routine for visualization.

The first part corresponds to the definition of an array of data-containers whose configuration is the same as the actual sample libraries. In the second part, experimental data are extracted from the files and each of them is linked with the corresponding data-container according to their coordinates in the sample library. After finishing these procedures, individual data can be accessed by tracing the tree structure; for example, the series of wavelength values in Fig. 2, which is pointed by the black arrow, is referred by the *Ruby* sentence displayed in the "Code" box, which is located at the second line in the window. The word "`vl`" at the beginning of the sentence means a variable corresponding to a VSL and being defined in advance by another *Ruby* sentence shown at the first line of the window, where three data files are specified in the parenthesis to construct the VSL.

Visualization is performed by calling an optional graphics library, Ruby/PGPLOT[5].

4. RESULTS

The appearance of the glass segments as a function of the position in the library and/or the annealing temperatures are plotted in Fig. 3. The glass segments named $A_n (n = 0, 1, 2)$ and $B_n (n = 0, 1)$ are located in the same positions in the three sample libraries, respectively. The annealing conditions for these segments are summarized

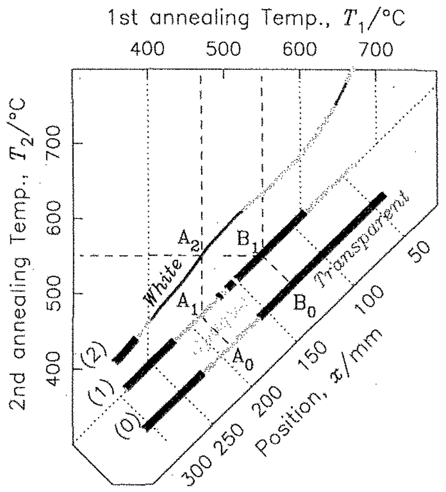

Fig. 3: Annealing condition of the sample libraries and appearance of the annealed glass segments inside. (0) with no annealing treatment for reference. (1) with 1st annealing for 5min only. (2) with two successive heat treatments, each for 5min. Thick black line: completely transparent segment, thin black line: white, and medium gray line: opaque(see text). A_n and B_n indicate specific glass segments. These notations are also used in Fig. 4.

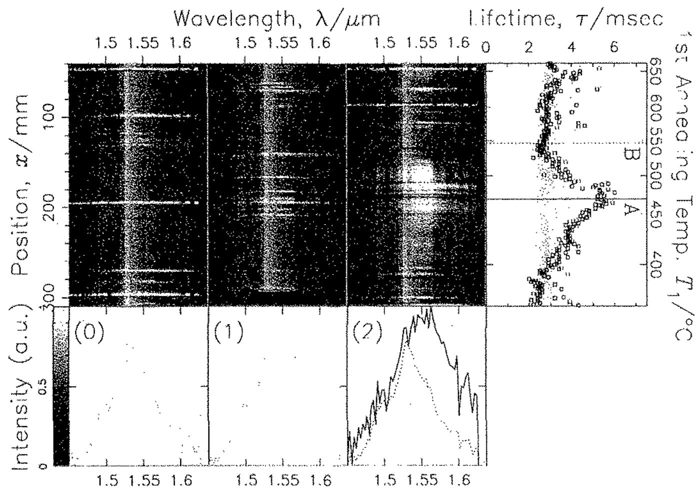

Fig. 4: Fluorescence spectra of Er^{3+} and lifetime of 1.533nm fluorescence plotted along the library or as a function of the 1st annealing temperature, T_1(see text). Excitation wavelength is 977nm.

by the following.

$$\text{(melt)} \xrightarrow[T_1 = 470]{\text{1st anneal for 5min}} \; \xrightarrow[T_2 = 550]{\text{2nd anneal for 5min}} \quad (1)$$
$$\downarrow A_0 \qquad \downarrow A_1 \qquad \downarrow A_2$$

$$\text{(melt)} \xrightarrow[T_1 = 550]{\text{1st anneal for 5min}} \quad (2)$$
$$\downarrow B_0 \qquad \downarrow B_1$$

As the heat treatment proceeded, crystallization occurred and the portion of transparent segments in the library decreased.

The remainder of the multi-dimensional data are shown in Fig. 4. Three contour charts in gray scale located at the top left are the positional dependence of fluorescence spectra, whose sectional views at positions of A and B are shown at the bottom right in thick and broken line, respectively. While no positional dependence are observed for the libraries of (0) and (1), the glass segments around A_2 shows some spectral broadening. At the same time, an increase of fluorescence lifetime is observed at around A_2 as shown at the top right of Fig. 4, where the black points correspond to the library (2) and the gray dots to the library (0) and (1).

Consequently, the segment of A_2 shows the highest fluorescence lifetime. This is probably because Er^{3+} ions moved from the oxide glass matrix to the precipitated fluorine-rich phase during the 2nd heat treatment[4]. This phenomenon does not occur if the 1st heat treatment of 470°C is omitted such as B_1(see Eq. 2).

5. DISCUSSION

In order to make the benefit of VSL clear, let us consider the process of making Fig. 3 and 4 without using VSL. The experimental data used here are listed in Table 1. The annealing temperatures (#2 and #3 in Table 1) are functions of the position in the sample library (#1) and are calculated from the temperature gradient of the furnace and the annealing position of the glass capillary in the furnace.

In general, the temperature gradient of the furnace is not linear. Therefore, the conversion from #1 to #2 and/or #3 is not so simple. For this reason, the intervals of tick marks along the x-axis in Fig. 3 and T_1-axis in Fig. 4 are not fixed and the shape of the library (2) plotted in Fig. 3 is not straight. This means that this conversion is needed whenever we plot a graph in which both the x-axis and T_1-axis are included.

Without using VSL, Fig. 3 and 4 can be plotted only after merging #1–#4 data and #1, #2, #5–#7 data (see Table 1), respectively, which are stored separately in different format. Moreover, the conversion of $x \rightarrow T_1, T_2$ is needed every time we merge them. One of the advantages of VSL is to avoid this merging process, which is needed only once when the VSL is constructed.

6. CONCLUSIONS

Concept of virtual sample library (VSL) is proposed for managing multi-dimensional data obtained from several measurements on one combinatorially integrated sample library. VSL is a data medium for storing data hierarchically and accessing data intuitively. It reduces the load of merging experimental data which are separately stored in different formats. This is realized with the aid of abstract expressive power of object-oriented language. Plotting 7-dimensional data in two figures is demonstrated.

References

[1] "Ruby Home Page." (http://www.ruby-lang.org).

[2] D. Thomas and A. Hunt, *Programing Ruby, The Pragmatic Programmer's Guide*. Addison-Wesley, 2001. (ISBN 0-201-71089-7).

[3] C. Laird, "Open source in the lab," *IBM developerWorks™*, Oct. 2002. (http://www-106.ibm.com/developerworks/linux/library/l-oslab/ *or* http://www-6.ibm.com/jp/developerworks/linux/030110/j_l-oslab.html *in Japanese*).

[4] S. Todoroki and S. Inoue, "Combinatorial fluorescence lifetime measuring system for developing Er-doped transparent glass ceramics," *Appl. Surface Sci.*, 2003. in print.

[5] M. Tanaka, "Ruby/pgplot." http://www.ir.isas.ac.jp/~masa/ruby/pgplot/index.html.

Acknowledgments

The first author express his sincere thanks to Yukihiro Matsumoto (Ruby) and Masahiro Tanaka (NArray & Ruby/PGPLOT) for providing their excellent software for free.

(Received October 12, 2003; Accepted November 19, 2003)