

カラードットコードシステム

上田謙一†

(有) ADEU.NEK†

1. はじめに

文書や音声などの各種のデータを、カラードットコード画像として表現して、コンピュータ内部ではこの画像をベースにして基のデータを管理し、印刷する場合には、この画像を各ドットの大きさおよびドットとドットの間隔を制御して印刷し、印刷記録されたカラードット画像をスキャナやカメラで読み取り、コンピュータ内部で基のデータを復元する、「カラードットコードシステム」という手法を提案している。提案手法により、これまでのコンピュータ内部でのデータ管理の手法とは異なる、ファイル管理や各種のデータ管理、ソフトウェア開発管理などを実現することが出来る。

2. カラードットコードシステムの概要

文書や音声などの各種のデータを、例えば、印刷などの基本のインクの種類である、C(シアン)、M(マゼンタ)、Y(イエロー)、K(ブラック)等のカラーコード列で対応させて表現し、このカラーコード列に対応させたカラーのドット画像を生成し、生成されたカラードット画像をベースにして、基の各種のデータ管理、ファイル管理、ソフトウェア開発管理を行い、プリンタで印刷する場合には、この画像の各カラードットの大きさおよびドットとドットの間隔を制御して、紙などの媒体にコンパクトに印刷することにより、大規模データを記録し、スキャナやカメラで読み取って、基のデータを復元することにより、コンピュータ内部のデータと印刷記録されたデータとの連続性を実現する。

例として、「私は敬子です。」という文書のデータのカラーコード列は、図 1b として表され、対応するカラードット画像は、図 1c となる。次に、この画像を印刷する場合には、パソコン用のインクジェットプリンタでドットのサイズを、

例えば、0.1mm×0.1mm とし、ドットとドットの間隔を 0.1mm とすると、印刷されるカラードット画像のサイズは、2.8mm×0.8mm となる。

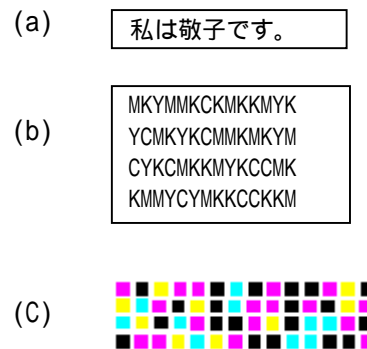


図 1. 文書データのカラードット画像

3. カラードット画像を基にしたデータ管理

このようにして、各種のデータをカラードット画像に変換し、この画像を用いて基のデータ管理を行うことを考える。このための代表的な例として、HTML 文書と、ソフトウェア開発におけるプログラム管理をここでは取り上げることとする。まず、HTML 文書では、文書内容について詳細を記述したい場合、ハイパーリンクにより、詳細を記述した別の文書ファイルを指定するという方式がもちいられる。この場合、この文書を印刷した場合、ハイパーリンクにより指定された別の文書情報は失われてしまう。これに対して、カラードット画像に基づく文書の場合は、図 2 に示すように、ハイパーリンクではなく、別の文書ファイルのデータの実体そのものをカラードット画像として埋め込んでしまうというものである。そして、コンピュータ内部では必要に応じてこのカラードット画像をマウスなどでクリックすることにより、基のデータが復元されて表示され、印刷する場合にはカラードット画像がそのまま印刷されるので、必要に応じてスキャナやカメラで読み取って、基のデータを復元することにより、印刷しても基のデータがそのまま保存されこととなり、コンピュータ内部のデータと印刷記録されるデータと

の連続性が保たれることとなる。

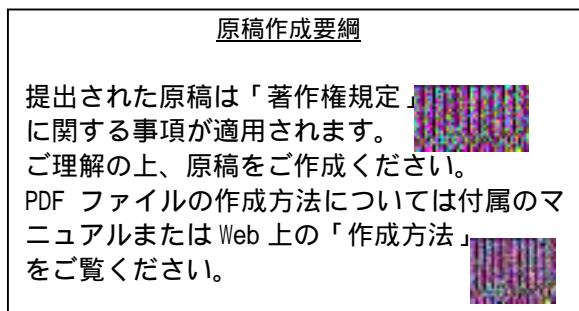


図 2. カラードット画像を埋め込んだ文書

次に、ソフトウェア開発管理におけるプログラム管理をカラードット画像を用いて行うことを考える。通常、プログラムはいくつかのサブプログラムから構成され、各サブプログラムごとに開発者が異なるというのが一般的であり、この場合、プログラムのバージョン管理が重要となる。このようなプログラム管理をカラードット画像を用いて行うと、以下のようになる。

まず、画像を貼り付けるために一つのキャンバス（テンプレート）を用意する。そして、このキャンバスにサブプログラムのカラードット画像を、図 3 のように貼り付ける。カラードット画像は、任意の形状にすることが出来るので、サブプログラムごとに円とか三角とか正方形にするなど、形状を変えることによって、類似のサブプログラムを同じ形状にするなどの方法を取ることが出来、全体としてのプログラムの識別力を向上させることが出来る。

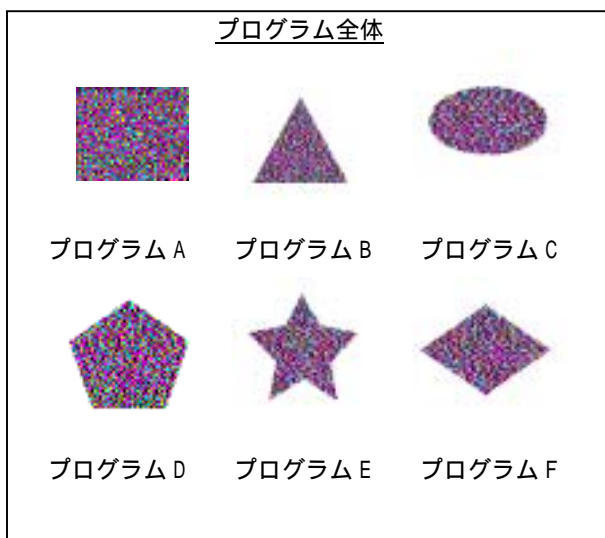


図 3. サブプログラムをカラードット画像として貼り付けたプログラムキャンバス

これにより、サブプログラムのうちの一つを更新した場合、その更新の日付がキャンバス自身が更新された日付ということになり、また、一つのサブプログラムの更新の影響がどこまで及ぶのかという事の管理が目で見えてすぐわかることとなる。また、別のプログラムキャンバスにサブプログラムの画像をコピーして貼り付けることにより、同一のサブプログラムがどのようなプログラムに共通に利用されているのか、ということが可視化される。更に、アプリケーションシステムの機能ブロック図にプログラムオブジェクト自身をカラードット画像として貼り付けることにより、機能仕様書とプログラムを一体化させたアプリケーション開発管理が実現されることとなる。さらに、カラードット画像として、プログラムとデータ、あるいは、ソースリストとオブジェクトコードという、性質の異なるオブジェクトを同一のキャンバスに貼り付けて全体の一元管理を行うことが可能となる。

5. おわりに

本稿では、文書や表や音声などのあらゆるデータをカラードット画像として表現し、コンピュータ内部では、この画像をベースにして基のデータの管理を行い、印刷する場合には、この画像の各ドットの大きさおよび、ドットとドットの間隔を制御して印刷し、印刷記録されたカラードット画像をスキャナやカメラで読み込んで基のデータを復元するという、カラードットコードシステムという手法を提案した。そして、カラードット画像でデータ管理を行う方法の例として、ハイパーリンクに基づく HTML 文書とは異なる、実体を本文に埋め込んでしまう方法と、仕様書とオブジェクトを一体化させたソフトウェアプログラム管理について記述した。今後は、OS が、必要に応じて各種のデータ（オブジェクト）をカラードット画像に変換して管理し、変換された画像を復元するという機能を基本機能として持たせることを目指す。

参考文献

- [1] 上田謙一：” カラードットコードシステム ” 画像電子学会研究会予稿 03-05-11 pp63-69
- [2] 上田謙一：” カラードットコードシステム ” エレクトログラフィック特別研究会予稿 2004 年 2 月
- [3] 特願 2003-108382
- [4] <http://www.asahi-net.or.jp/~EA9K-UED/ADEU.NEK/>