Quality and Architecture Issues Related to the Development of Internet-Based Image Fulfillment Servers

Robert John Parada, Jr.
Eastman Kodak Company
Rochester, New York

Abstract

Basic architectural issues related to the development of Internet-based image fulfillment services are presented. The interface requirements and constraints under which such services must operate are described, including platform independence, standard transfer protocols, the specification of a desired product, and the handling of device-specific parameters/settings. The image science needs of these applications are discussed, with emphases on the importance of standards for image characteristics, such as color encoding, spatial information, the capture process, etc., and on the responsibility for, and limitations on, image quality. A system architecture that would satisfy the primary needs and constraints of Internet-based imaging servers is hypothesized. This architecture involves the use of an eXtensible Markup Language (XML). Completed by the World Wide Web Consortium (W3C) in early 1998, the XML standard allows data to be self-describing and is a ready-made solution for Internet communication. Its use would provide an efficient, extensible means for the development of diverse Internet-based imaging applications.

Introduction

The growth of e-commerce, the proliferation of image digitization services, and the increased affordability of digital capture devices, such as digital cameras and flatbed scanners, have all contributed to the growth of Internet imaging applications and services. As the number of digital images continues to rise, the market for the production of digital service prints and specialty products can be expected to grow. This will lead to the need for sophisticated image fulfillment servers.

An image fulfillment server is an Internet-accessible application that receives and processes requests for the generation of digital and/or hardcopy products from digital inputs. Example products include CD image archives, standard-format silver halide prints, and specialty products like coffee mugs and mouse pads. Such servers compose the "back end" of Internet-based imaging services. Given the openness and accessibility of the Internet, these servers may be expected to fulfill requests from a variety of "front end" applications. The associated interchange of data between these applications creates a number of interesting challenges, including the need to segregate processing responsibilities and to define an unambiguous data interface. Standards for data formats, image characteristics, and processing instructions are important pieces of this interface definition.

As for any Internet-based application, it is desirable for an image fulfillment server to contain a set of web-friendly features. These features include modularity, extensibility, platform independence, transmission efficiency, and compatibility with the existing Internet architecture.[1,2]

In the following sections, quality and standards issues related to the development of image fulfillment servers are described, alternative server architectures are compared, and the possible advantages of using an XML-based syntax to communicate image processing instructions is discussed.

Quality and Standards

There is general agreement in the photofinishing community concerning consumer expectations on single-stimulus product quality for *traditional* (optical) photofinishing.[3] In cases where the consumer already has a print in hand and requests a reprint, the tolerances on the reproduction process become tighter, because the consumer desires a reprint to match the "true" (i.e., original) print. Internet-based fulfillment servers must ideally operate under these tighter tolerances, because they often serve as a convenient means of generating digital "reprints" of optical originals. In practice, the appropriate tolerances depend on consumer expectations for *digital* photofinishing. Recent studies suggest that some consumers will not differentiate their expectations based on whether the original image was captured optically and later digitized (e.g., from a negative or print), or was captured digitally via a digital camera.[4] However, further studies of consumer expectations for traditional and digital fulfillment are needed to better understand the differences between them.

Understanding consumer expectations will help to formulate guidelines and recommendations on the remote fulfillment of digital images. To minimize dissatisfaction, the consumer could be given clear warnings about fulfillment options and editing choices that would likely result in inferior-quality products.

Quality issues related to remote fulfillment extend beyond the fulfillment server. The capture/digitization process may significantly limit the inherent quality of a given digital image. Examples include the use of inferior scanning equipment to digitize color negatives and the use of images captured by low-resolution digital cameras to create large-format prints. Softcopy-based previewing/ editing can cause further problems. Display hardware limitations (e.g., resolution and color gamut), improper settings (e.g., color temperature, gamma, and color balance), and environmental conditions (e.g., illuminant type and flare level) may invalidate such predictions and lead to consumer confusion. Unfortunately, the consumer's preview hardware and environment normally cannot be tightly regulated to avoid such problems.

Ultimately, the service provider may be held responsible for inferior-quality products, even if such quality issues were not the fault of the fulfillment server. Therefore, despite the associated difficulties, it is desirable to implement quality safeguards. When safeguards exist at all, they are often limited to checks on image resolution, i.e., to ensure there are an adequate number of pixels present to create the requested product. These simplistic mechanisms have limited usefulness. They do little to advise a consumer that an image is improperly balanced, or that it contains substantial compression artifacts. Further, their implementation often varies from one service provider to another. The standardization of quality warnings would help to send clear, consistent messages to the consumer about the fulfillment limitations of a particular image.

Internet-based imaging applications also benefit from standards in areas other than quality. Standards are a fundamental cornerstone on which these applications must be built, and there are a number of distinct layers that are required. The most important layers include data format, image characteristics, and image processing.

Data format standards cover basic connectivity concerns, including order/product specifications, image file formats, metadata encapsulations, and transfer protocols. A number of applicable standards are already used by networked applications. For example, the transfer protocol standards ftp and http have been in use for years. Image formats commonly supported by Internet applications include the Tagged Image File Format (TIFF) and the JPEG File Interchange Format (JFIF). Recently, the EXtended Image Format (EXIF) has received particular attention because of its lossy-compression and metadata-storage capabilities.[5]

Order/product specifications have not yet been well standardized. The Photofinishing Data Format (PfDF) has begun to gain acceptance in the digital photofinishing community.[6] This format allows for control and tracking of batch, order, and frame information within a photofinishing lab; it was not designed with the transmission order information between Internet-based applications in mind. The Digital Print Order Format (DPOF) has been endorsed by several major imaging companies for use in the fulfillment of digital camera images.[7] This standard provides a mechanism for exchanging order information and basic formatting instructions, but is limited in scope to the straight-forward fulfillment of digital camera images. At present, there is no extensible, widely used industry standard for the encapsulation and exchange of digital fulfillment order information between Internet-based applications.

Though frequently overlooked, standards used to specify the inherent characteristics of a digital image are also of critical importance for net-based fulfillment servers. These standards are just now being established. For example, the sRGB color space is now widely excepted as the default color encoding for the Internet.[8-10] This standard is appropriate for many situations in which digital images are exchanged between multimedia systems—particularly when the images are to be viewed on typical video-RGB displays. In some situations, the bit-depth and gamut limitations of this color space may unnecessarily limit the reproduction capabilities of certain high-quality, hardcopy output devices and/or the extent to which color and density characteristics can be adjusted. Fulfillment servers may, therefore, need to consider support of additional standard color spaces.[11]

For many image characteristics, no widely accepted standards exist yet. Consider the image structure characteristics of sharpness and noise. To date, these characteristics have been largely unregulated. This leads to ambiguity as to how an image fulfillment server should process an image to produce a digital product of "optimum" quality. (The definition of "optimum" is left open to the interpretation of the reader). The development of such standards would, therefore, be of significant benefit to net-based imaging applications. In the interim, proper use of available image-file metadata tags is crucial.

Even if the data format and image characteristics are known precisely, in order to produce the desired product, net-based applications must be able to effectively communicate the image processing operations desired by the consumer. Consumer wishes concerning image orientation, text annotation, color balance, etc., must all be conveyed to the fulfillment server if this application will be responsible for the implementation of these processing steps. A standard means of specifying processing instructions would simplify this issue. This specification must implicitly convey knowledge of how to implement the instructions, i.e., via specific algorithms. Otherwise, instructions may be misinterpreted. For example, several standard algorithms can be used to resample an image. The use of the wrong technique may perceptibly alter the appearance of the processed image.

Without each of the specified layers of standards, there will exist the potential for miscommunication of image data and processing intentions between net-based applications. Only through their continued development will unambiguous protocols be established. The development of standards for image processing instructions, in particular, is still fairly nascent. The option of using of a text-based mark-up language to develop such a standard is discussed later.

Alternative Processing Architectures

In its simplest form, Internet-based imaging can be broken down into a "system" consisting of two basic components: (1) a *front-end* or "sending" application that initiates an order, and (2) a *back-end* application (i.e., fulfillment server) that receives and completes the order.

In one possible architecture for Internet imaging, the bulk of the processing is handled by the front-end application. This *decentralized* architecture, so called because the majority of the processing is implemented by individual front-end applications rather than by the fulfillment server, is depicted in Figure 1.

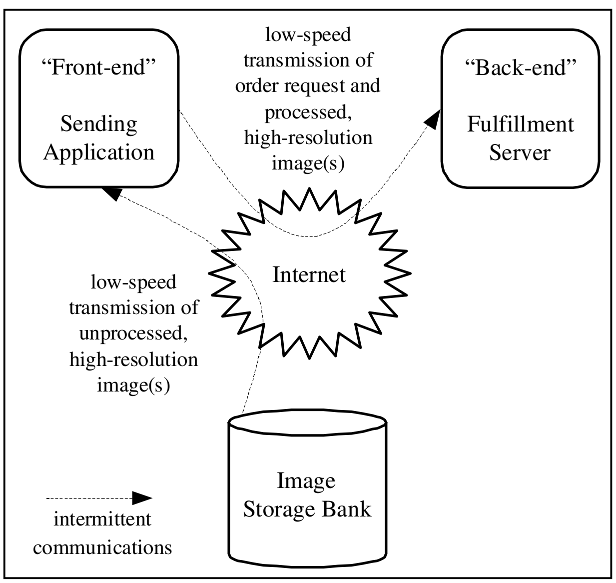

Figure 1. Decentralized architecture for Internet imaging

In the decentralized architecture, consumer imagery can either be resident on the "front end" processor or stored in an Internet-accessible image storage bank. In a typical use scenario, a consumer would download a high-resolution copy of the image(s) to the local machine running the front-end application. The image(s) would then be enhanced/edited and formatted to create the desired product. Once satisfied with the results, an order request and the processed, high-resolution product would be transmitted to a fulfillment server. The server would complete any remaining processing required to create the final product.

The decentralized architecture is used today by many electronic imaging products. These front-end applications send processed files to a fulfillment service for order completion. Representative applications include Microsoft PictureIt®, Adobe PhotoShop® and PhotoDeluxe®, MGI PhotoSuite®, and Kodak Picture Easy®.

There are several advantages to a decentralized scheme. For one thing, the flow of information is very simple: data is transmitted from the front-end application to the fulfillment server. The consumer can be provided with a product preview to give him/her the opportunity to accept or revise these operations before the order is transmitted. For another, implementing the bulk of the processing in the front-end application means that these steps do not need to be communicated to the fulfillment server. However, the fulfillment server will ordinarily be required to carry out a certain amount of additional processing. For example, proper color management, resampling, and sharpness adjustments all require integral knowledge of the imaging characteristics and configuration of the output device that will create the final product.

The primary disadvantages to this architecture involve processing and transmission efficiencies. The computing capabilities of the local machine (e.g., a consumer's personal computer) will often be inferior to those of a commercial fulfillment server. Coupled with the fact that all processing must be performed on high-resolution images, this means that the processing efficiency of this scheme is relatively poor. Typically, images having dimensions of 1.5 megapixels (e.g., 1000 pixels by 1500 pixels) or more are required to produce high-quality digital prints at standard sizes. Current modem communication rates make the transfer of such large-dimensioned images rather tedious. Optionally, some amount of lossy compression can be applied prior to transfer, but caution must be taken to prevent appreciable degradations in image quality.

An alternative to the centralized scheme is to implement most/all of the processing at the fulfillment server. This *centralized* architecture, so called because the processing is implemented by a common server rather than by separate front-end applications, is depicted in Figure 2.

In the centralized architecture, consumer imagery is best stored in an Internet-accessible image storage bank at two or more resolutions. In addition to consumer images, this bank can also be a repository for creative borders and other graphics. In a typical use scenario, a consumer would download a low-resolution version of the desired image(s) to the local machine running the front-end application. The image(s) would then be enhanced/edited and formatted to create a virtual product. Once satisfied with the results, an order request consisting of processing instructions and the location(s) of the corresponding high-resolution image(s) in the image bank would be sent to the fulfillment server. The server would then download the required high-resolution image(s) through a high-speed communications channel and execute the processing instructions to create the desired product.

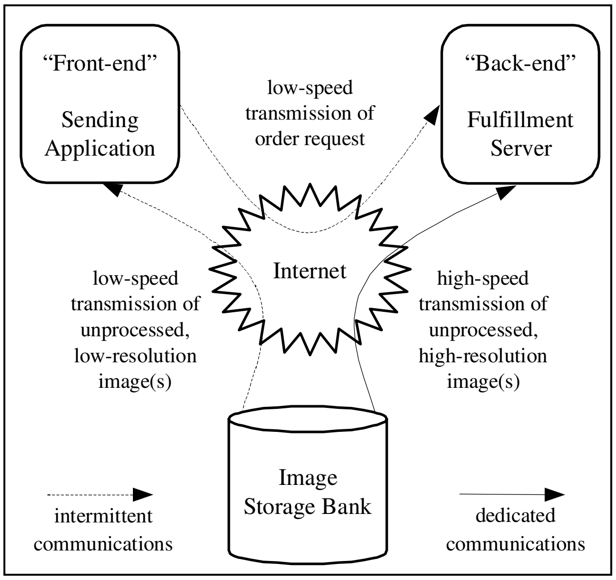

Figure 2. Centralized architecture for Internet imaging

To date, the use of a completely centralized architecture is rare. Nevertheless, it is advantageous in situations where the majority of images reside in a centralized location. Examples include on-line storage archives of consumer imagery and professional photofinishers who wish to allow customers to remotely proof, select, and order their images.

A centralized architecture can be much more efficient than a decentralized scheme. Only low-resolution images need be transferred across low-speed communications lines; higher speed, dedicated communications can be established between the image storage bank(s) and the fulfillment server. Furthermore, only low-resolution images need be processed on front-end machines; the more-powerful fulfillment server processes all high-resolution imagery.

A centralized scheme can also help to minimize artifacts resulting from unnecessary processing steps. For example, the creation of composite products via a decentralized architecture may require two resampling stages: one by the front-end application to format and composite the images, and a second at the fulfillment server to adjust the composite dimensions for the appropriate output device. A properly architected centralized scheme can implement all required resizing via a single operation, because the server conducts all processing and knows the configuration of the output device. The direct transfer of high-resolution data between image bank and fulfillment server minimizes the opportunity for an image to undergo multiple stages of compression, which can also lead to objectionable artifacts.

Despite the above advantages, centralized architectures are not without their own challenges. Principal among these is the fact that the processing completed by the front-end application needs to be replicated by the fulfillment server. This implies that the desired processing must be effectively communicated between the two applications and that they must have equivalent processing capabilities. A standard set of instructions and related algorithms is therefore required, along with a means of version control to keep the two applications synchronized. Without these standards, the creation of complex products (e.g., multi-image composites) would be very difficult to implement via this architecture.

It warrants mention that neither of the described architectures alleviates the problems inherent with inferior digitization/capture or the use of softcopy preview to predict final-product quality. These problems must be overcome via proper hardware selection and calibration/ characterization, establishment of viewing environment recommendations, and consumer education. While these topics are certainly important to the future of Internet imaging, they are not given further attention here.

The remainder of this paper is dedicated to one particular option for developing a standard specification for image processing instructions. As stated above, such a standard is a crucial component of a robust centralized architecture for fulfillment servers.

XML and Internet Imaging

Completed by the World Wide Web Consortium (W3C) in early 1998, the eXtended Markup Language (XML) standard allows data to be self-describing and is a ready-made solution for Internet communication.[12] XML, like HTML, is a text-based means of specifying the formatting and layout of data. It has the additional capability of allowing data to be tagged so that its purpose is understood.[13,14]

As its name implies, XML is an extensible language. Data definitions may be customized to fit a given application. The use of XML has already been investigated for a variety of fields, including astronomy, banking, chemistry, genetics, health care, journalism, mathematics, meteorology, and music. Although theorizing about the specifics of a working XML data dictionary for Internet imaging is beyond the scope of this paper, it is useful to discuss how the use of the XML syntax could fit into the centralized architecture described earlier (see Figure 2).

An XML-based standard for image processing instructions would allow explicit processing instructions to be passed between a front-end application and a fulfillment server via a simple text file, i.e., a file conforming to the XML syntax. This approach would be simple, extensible, and platform-independent. Furthermore, it would maintain relatively high transfer efficiencies, would support features like order validation and internationalization/localization, and would maintain compatibility with existing Internet protocols.

Encapsulating processing instructions in an XML file would effectively partition the instruction syntax from the actual implementation. Consequently, a sufficiently generic set of instruction definitions could be used by a number of different processing engines. Practically, this means that products/services could differentiate themselves by using proprietary processing algorithms. A given service provider might be able to advantage his/her products through superior color balancing, noise suppression, annotation features, etc. The caveat to this is that there must still be a common understanding between front-end applications and a given fulfillment server as to how the instructions will be implemented. In other words, the same processing library must be used by both applications. With this caveat in mind, there may be advantages to developing industry standards for a set of common processing algorithms (e.g., resampling, cropping, rotation, etc.). The use of proprietary algorithms for these and other operations could remain an option.

It is important to recognize the limitations of XML. Its use would merely provide a mechanism by which to specify *explicit* processing desires. Careful thought should be given about how to handle "smart" server-side algorithms, i.e., algorithms implemented by a fulfillment server that analyze an image to determine how to customize processing for it. Processing steps implemented by a fulfillment server that were not explicitly requested and previewed at the time of order submission may detrimentally change the acceptability of the processed image data.

Furthermore, XML does not permit the transmission of parameters commonly used to complete processing. The transmission of complex Look-Up Tables (LUTs), ICC profiles, and so on must be handled via some other mechanism.

Finally, as discussed earlier, a fulfillment server ordinarily must implement a number of processing steps to properly format the image data for output. Without intimate knowledge of the fulfillment equipment, the front-end application cannot include these instructions in the XML file it creates and passes to the back-end application. The use of XML, therefore, does nothing to alleviate the need for the fulfillment server to handle such steps.

Each of the limitations described above can be overcome through careful system design. The use of XML to develop an industry standard for specifying processing instructions, therefore, remains a tempting notion.

Conclusion

Major interface requirements for networked image fulfillment servers and the constraints under which these applications must operate have been presented. Image quality issues affecting these servers have been discussed, along with the need for standards on quality, data formats, image characteristics, and processing instructions. Two system architectures have been described. Special attention has been given to an efficient, *centralized* architecture that implements server-side processing. An overview of the benefits and limitations of using XML as a component of this architecture has been given. It is felt that using XML to develop an industry standard for specifying processing instructions would benefit the development of diverse, efficient Internet-based imaging applications.

References

1. J. Bosak, XML, Java, and the future of the Web, Sun Microsystems, October 1998, http://sunsite.unc.edu/pub/sun-info/standards/xml/why/xmlapps.htm.
2. T. Berners-Lee, *Web Architecture from 50,000 feet*, September 1998, http://w3c.bilkent.edu.tr/DesignIssues/Architecture.html.
3. D. Schulmerich and H. Ikoma, "High-quality prints through neg/pos system enhancements," *IS&T 10th International Symposium on Photofinishing Technology*, 1998.
4. M. E. Miller and R. Segur, "Perceived image quality and acceptability of photographic prints originating from different resolution digital capture devices," *Proc. IS&T 1999 PICS Conference*, pp. 131-136, 1999.
5. JEIDA: *Digital Still Camera Image File Format Standard (Exchangeable image file format for Digital Still Cameras: Exif)*, Version 2.0, October 1997.
6. J. Bird, *Photofinishing Data Format (PfDF): Specification Document*, Version 4.x, 1999, http://www.pima.net/standards/ it13/it13_POW.htm.
7. *DPOF (Digital Print Order Format) Summary*, Oct. 1998, http://www.panasonic.co.jp/avc/video/dpof/index_e.htm.
8. IEC/3WD 61966-2.1: Multimedia systems and equipment - *Colour measurement and management – Part 2.1: Colour management - Default RGB colour space – sRGB*, , International Electrotechnical Commission, Technical, Jan. 1998.
9. M. Stokes et al., *A standard color space for the Internet – sRGB*, v1.10, November 1996, http://www.w3.org/Graphics/Color/sRGB.html.
10. M. Anderson et al., "Proposal for a standard default color space for the Internet – sRGB," *Proc. IS&T/SID 4th Color Imaging Conference*, pp. 238-246, 1996.
11. S. Süsstrunk, R. Buckley, and S. Swen, "Standard RGB Color Spaces," *Proc. IS&T/SID 7th Color Imaging Conference*, pp. 127-134, 1999.
12. *Extensible Markup Language (XML) 1.0*, T. Bray, J. Paoli, and C. M. Sperberg-McQueen, eds., W3C, February 1999, http://www.w3.org/TR/1998/REC-xml-19980210.
13. J. Bosak and T. Bray, "XML and the second-generation web," *Scientific American*, May 1999, http://www.sciam.com/ 1999/0599issue/0599bosak.html.
14. D. Connolly, "The XML revolution," *Nature Web Matters*, Oct. 1999, http://helix.nature.com/webmatters/xml/xml.html.

Biography

Robert Parada received his B.S. degree in imaging science from RIT in 1992, his Ph.D. degree in Optical Sciences from the University of Arizona in 1997, and his Ph.D. degree in Optical Physics from the Université du Littoral (France) in 1997. He has been a senior research scientist at Eastman Kodak Company since 1997. His work has primarily focused on the development of consumer imaging products, with emphases on digital photofinishing, Internet imaging, and general image quality.