

Strategies for effective use of intelligent graphics in Web applications

John C. Gebhardt

Micrografx, Inc., Annapolis, USA
johng@micrografx.com
<http://www.micrografx.com>

Greg Gallant

Micrografx, Inc., Annapolis, USA
gregg@micrografx.com
<http://www.micrografx.com>

Abstract:

No abstract was provided for this paper.

Introduction

Graphics are essential content elements in many Web documents and applications. In general, we would all agree that good graphics make Web applications easier to use and understand and provide very intuitive mechanisms for human users to interact with and control complex business processes and information. On the other hand, a very small fraction of the population has the artistic talent to design and create good graphics. In order to be able to effectively deploy highly visual and interactive Web applications economically, we need to develop strategies for managing intelligent graphical content that maximize the efficiency of our very scarce and precious artistic resources so we can economically archive the high levels of utility and ease-of-use in our Web applications that we all are striving for yet seldom achieve.

Fortunately, new, Web standards such as WebCGM and SVG make it possible to create and deploy intelligent, scalable graphics in Web applications in open, non-proprietary environments. As tools emerge which implement these new standards, new opportunities to use graphics in new and exciting ways are appearing all the time. The emergence of these new graphics standards and technologies has made it possible to create graphics-intensive Web documents and applications which do not consume the large chunks of bandwidth typical of Web Pages which use traditional image-based graphics.

This paper explores various strategies for deploying graphics-intensive documents in Web-based applications using existing and emerging W3C graphics standards. The raster graphics formats currently in widespread use on the Web such as GIF, JPEG, and PNG have limited capability for scaling and interactivity. The new standards such as WebCGM and SVG are much more flexible and versatile, giving applications architects many more options for using graphics in their applications.

For example, the ability to zoom, and pan a WebCGM or SVG graphic make it possible to use much more detailed and complex drawings in Web-based applications. The new graphics standards also provide the capability to attach hyperlinks and other metadata to pieces of a graphic. How the hyperlinks and metadata are authored and managed is a critical issue.

Finally, some of the issues surrounding the creation and management of graphics for Web applications are discussed. Because graphics are expensive to create and maintain, and not all browser implementations support graphics standards uniformly across all platforms, graphics reusability and interoperability are high priority issues. The paper discusses some of the techniques and trade-offs for creating and managing intelligent graphics so that graphics-rich applications can be economically deployed in today's Web environments.

What are "Intelligent Graphics"?

"Intelligent Graphics" as graphics that contain graphical objects and associated metadata that cause the graphical objects to be responsive to user-generated or external events. A typical response might be that a graphical object changes color and displays a message identifying itself when a user "points" to it on a display screen. If the user "clicks" the pointing device while pointing to the object, it could present the user with a menu of choices of possible hyperlinks to execute. Another typical response might be that one or more graphical objects moves from the "assembled" position relative to other objects in the graphic to the "disassembled" position in response to the user "pushing" (by clicking) a graphic which depicts a push button labeled "Disassemble".

Intelligent Graphics are typically made up of graphical objects – images, lines, arcs, curves, text, symbols, paths, groups of objects, overlays, etc.; attributes which describe how the graphical objects are to be presented – line weight, line type, text font, fill color, etc.; composition rules which describe a coordinate system and the relative layout of the objects; and metadata which describes how the graphic will respond to user-generated or externally generated events.

A typical intelligent graphic which might be integrated into an e-commerce service, support, and spare parts procurement system is shown in Figure 1.

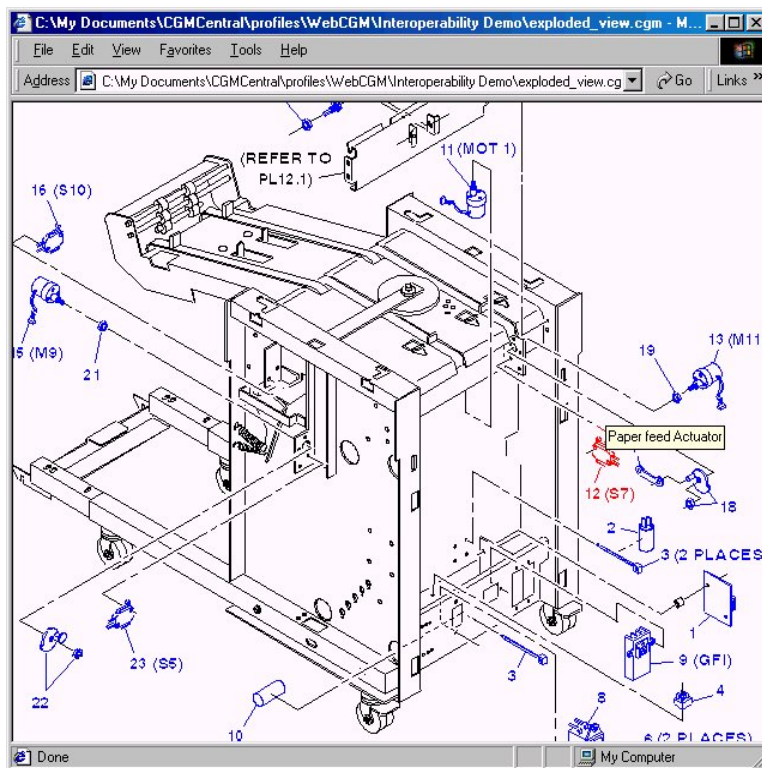


Figure 1. Typical intelligent graphic used in an illustrated parts catalog

There are several different ways to represent intelligent graphics for use in Web applications. There are four important open, non-proprietary Web graphics standards and several proprietary ones which are in use or advanced development as this paper is written. We will briefly discuss the capabilities of each, concentrating on the open standards.

Open Web graphics standards

1. PNG (Portable Network Graphics) – PNG is an open, extensible file format for the lossless, portable, compressed storage of raster images. It has been a Recommendation of the W3C since 1996. Indexed-color, grayscale, and truecolor images are supported, as well as an optional alpha channel. Sample depths range from 1 to 16 bits. PNG is designed to work well in Web applications. As such, it is fully streamable and images can be created in PNG so they can be progressively displayed. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors. Also, PNG can store gamma and chromaticity data for improved color matching on heterogeneous platforms. PNG is supported by many browsers and image editing tools. While PNG has a mechanism for adding descriptive text data to images, it has no built-in means for adding intelligent metadata such as would be required for defining active areas or "hot spots" on the image. HTML server-side and client-side image maps are the only mechanism for adding "intelligence" to PNG images.
2. JPEG (Joint Photographic Experts Group) is a lossy compression method standardized by ISO in 1990. The file format, commonly referred to as "JPEG" is actually JPEG JFIF, which was created by the IJG (Independent JPEG Group) for the exchange of single JPEG-compressed images. JPEG JFIF was released in October 1991 and images in this format are widely used on the Web. The amount of compression can be adjusted to achieve the desired trade-off between file size and visual quality. Progressive JPEG is a means of reordering the information so that, after only a small part has been downloaded, a hazy view of the entire image is presented rather than a crisp view of just a small part. It is part of the original JPEG specification, but was not implemented in Web browsers until approximately 1996. It is now fairly widely supported. Plain vanilla JPEG is in the public domain. JPEG JFIF, like PNG, has no built-in means for adding intelligent metadata. Intelligent graphics applications must rely on HTML server-side and client-side image maps to add "intelligence" to JPEG images.
3. WebCGM – WebCGM is an "intelligent graphics" profile of the ISO CGM standard (ISO/IEC 8632:1999). It has been a Recommendation of the W3C since January 1999. In addition to a fairly rich complement of vector and raster graphical elements, WebCGM defines the semantics of non-graphical content (metadata) based on the Version 4 CGM elements, Application Structures and Application Structure Attributes. The non-graphical content allows the definition of hierarchies of graphical objects, as well as metadata, such as link specifications and layer definitions, associated with the objects. CGMs compliant with the WebCGM profile are binary files which meet the applicable W3C requirements for a scalable graphics format. The graphical content of WebCGM balances graphical expressive power, and simplicity and ease of implementation. The Air Transport Association's popular GREXCHANGE profile was taken as the starting point for the graphical content and modified only where it did not meet the requirements articulated in the W3C requirements. The graphical content of WebCGM is a simplified, but powerful subset of the ATA profile. A small but powerful set of metadata elements is standardized in WebCGM, to support hyperlinking and document navigation, picture structuring, layering, as well as search and query on WebCGM picture content. A particularly attractive feature of WebCGM is that many industry and government groups call out CGM as the preferred format for 2D technical graphics. Also, hundreds of applications can currently produce CGM graphics from which intelligent WebCGMs can easily be created. Figure 1, for example, was created on a CAD system as a CGM to which the WebCGM metadata was added.
4. SVG (Scalable Vector Graphics) – SVG is an XML-based language for describing two-dimensional graphics. SVG defines how to represent three types of graphic objects: vector graphic shapes (e.g., paths consisting of straight lines and curves), images and text. Graphical objects can be grouped, styled, transformed and composited into previously rendered objects. Text can be in any XML

namespace suitable to the application, which enhances searchability and accessibility of the SVG graphics. Nested transformations, clipping paths, alpha masks, filter effects, template objects and extensibility are some of the more interesting features of SVG. SVG drawings can be made intelligent, i.e. dynamic and interactive. The DOM (Document Object Model) for SVG, which includes the full XML DOM, and support for the new SMIL Animation framework allows for straightforward and efficient vector graphics animation. A rich set of event handlers can be assigned to any SVG graphical object. Because of its compatibility and leveraging of other Web standards, features like scripting can be done on SVG elements and other XML elements from different namespaces simultaneously within the same Web page. At the time this paper was written, the SVG specification was out for its second "final call". It is a work in progress; and will probably change during this period of public comment and feedback. In the meantime, several SVG rendering implementations are in various stages of development .

Other Web graphics standards

1. ActiveCGM – ActiveCGM™ is a profile of the International CGM standard for publishing intelligent, animated technical graphics in Web-based or other electronic publishing applications. It was developed by our Company, Micrografx, Inc. Micrografx markets software products for creating, editing and viewing ActiveCGM graphics. The ActiveCGM viewers work with Internet browsers and other applications. ActiveCGM graphics are typically used in Interactive Electronic Technical Manuals and Illustrated Parts Catalogs for technical illustrations, animated assemblies, and wiring diagrams. ActiveCGM graphics are also used as intuitive graphical user interfaces to access and present technical information stored in databases.
2. VML (Vector Markup Language) – Like SVG, VML is an XML-based language for describing two-dimensional vector graphics information together with additional markup which describes how that information is to be displayed and edited. VML was designed by Microsoft and submitted for consideration by the W3C as a candidate for the Scalable Vector Graphics specification. VML supports the markup of vector graphic information in the same way that HTML supports the markup of textual information. In VML, the content is composed of paths described using connected lines and curves. The markup gives semantic and presentation information for the paths. VML depends on CSS, Level 2 to determine the layout of the vector graphics which it contains. Like HTML, or most other XML vocabularies, VML describes objects which are intended to be subject to further editing. The objects which can be defined in VML are shapes or collections of shapes known as groups. VML is designed so that different editors can recognize and correctly handle each other's private data. Using liberal amounts of application-specific markup, interspersed with some VML, the latest versions of Microsoft's Office products can produce and interpret VML graphics.
3. Flash (SWF) – The SWF file format was designed by Macromedia from the ground up to deliver graphics and animation over the Internet. In 1998, Macromedia published the Flash format and began licensing Flash player source code to third parties. The SWF file format was designed specifically as a final-form delivery format and it is therefore not useful for exchanging graphics between graphics editors. SWF supports anti-aliasing, fast rendering to a bitmap of any color format, animation and interactive buttons. The format has an extension mechanism so with new features can be added. SWF files are compressed and support incremental rendering through streaming.
4. The GIF (Graphics Interchange Format) – The GIF format was developed in the late 1980s by CompuServe. Their intention was to define a protocol for platform-independent on-line transmission and interchange of raster graphic data. The GIF is defined in terms of blocks and sub-blocks which contain relevant parameters and data used in the reproduction of a graphic. A GIF Data Stream is a

sequence of protocol blocks and sub-blocks representing a collection of images. By specifying how long each particular image is to be displayed using control blocks, GIF files can be used for animations. A major issue in the use of GIF in Web applications is that it relies on the patented LZW compression technique.

Creating and managing intelligent graphics

The creation of intelligent graphics for Web applications can be approached in several different ways using a variety of tools and any of the standards described above. The best approach to use depends on a variety of factors. The most important factor is to determine the role of the graphic in the context of the overall Web application. This factor alone will often drive the choice of creation technique and presentation format. We will describe several common scenarios where intelligent graphics could be used and describe appropriate techniques and formats for each situation.

Hand-crafted intelligent graphics

In many cases, the role of the graphic is merely to draw interest to the page, such as an advertisement. If successful, the graphical content will help attract traffic to the site. The potential to generate direct and indirect consumer revenue often justifies the investment in creative talent and the creation of a one-off, unique presentation, including sophisticated animations. If the graphic is intended to be presented to a wide audience, the ability of mainstream Web browsers such as Microsoft's Internet Explorer and Netscape's Navigator to render the graphic must be seriously considered. Casual Web surfers are often reluctant to download ActiveX Controls, Netscape Plug-ins, or Java applets in order to be able to view content. For these consumer-focused, advertising applications, the best strategy is usually to use a good graphics design/animation tool to hand-craft a series of raster images which, when played in sequence, will produce the desired animated effects. A logical choice for a format to represent these graphics is the GIF format since virtually all browsers support GIF rendering natively. The major drawback of using GIF is that GIF files are bandwidth hogs and animated GIFs compound the problem. The designer must make the difficult choice between visual impact and download time. Usually, the choice is made in favor of minimizing download time and designers opt for small, carefully crafted, animated images with the absolute minimum number of frames necessary to convey a realistic sense of motion.

Hand-crafting large, animated, interactive pages - the Web equivalent of "full-page ads" - that load in a reasonable amount of time, is technically feasible using tools and one of the newer vector or hybrid formats such as SWF, VML, or SVG. However, the ability to view and interact with the page is dependent on which version of which browser is installed and whether the appropriate control, plug-in, or Java virtual machine has been installed. If and when SVG is adopted by the mainstream browser manufacturers, it will be a good choice for representing this type of page. In the meantime, SWF is being used extensively to design and craft interactive pages with high visual impact.

For sites designed to be used on a regular basis by relatively sophisticated users, the necessity to carefully configure the browsing environment to match the content is an acceptable inconvenience, at least in the short term.

These hand-crafted pages combine graphical components and metadata into monolithic functional units. They work well and can have high visual impact, but in general, hand-crafted graphics are expensive to create, consume scarce artistic resources, and are difficult to manage and maintain in the context of a

large, complex, e-commerce environment. If any change is necessary, the entire page must be edited and republished. This process is illustrated in Figure 2.

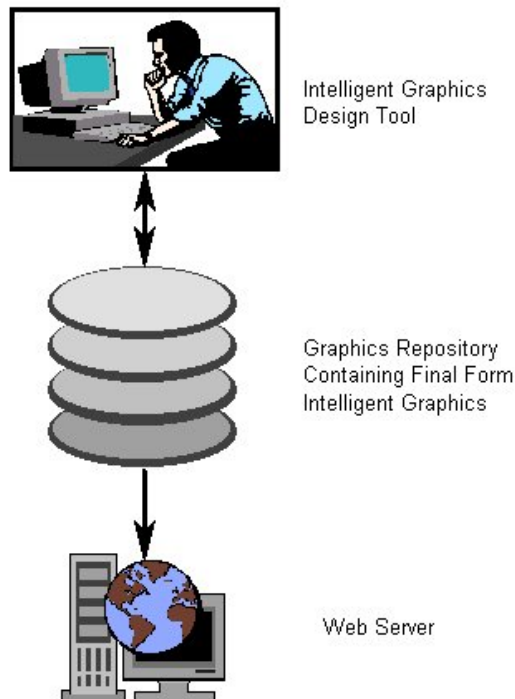


Figure 2. Hand-crafted intelligent graphics publishing process

Creating intelligent graphics from reusable components

In many e-commerce applications, the primary focus is to "Web-enable" existing business processes. Typical processes such as sales, procurement, customer support, service and repair are on-going operations in virtually every enterprise. These processes depend on large repositories of valuable information, including graphics. It is essential that as much of this existing legacy information be reused and repurposed as possible. Otherwise, the cost of deploying and managing e-commerce applications becomes prohibitive.

Turning a graphic produced for paper-based or other application into an intelligent graphic optimized for Web applications presents a number of challenges:

Legacy graphics challenges

Format diversity – legacy graphics are stored and maintained in a wide variety of formats, many of them proprietary. The good news is that most graphics tools can produce one or more of the standard Web formats. For example, most CAD systems can produce CGM files and most imaging tools can produce one of the popular Web formats such as PNG, JPEG, or GIF.

Lack of structure – legacy graphics seldom contain the structure necessary to create animations or hyperlinks. They are often raster images if they were scanned from paper or film. Even vector graphics are seldom structured into objects suitable for animation or hotspotting.

Complexity – Graphics intended to be used in traditional publications are often complex and detailed. They rely on the high resolution of traditional printing technology for the capability to pack a lot of information into page-sized images. For example, technical manuals often incorporate large-format foldouts for complex graphics such as schematics and wiring diagrams and mail order catalogs rely on high-quality, high resolution photos to attract and entice buyers.

Asynchronous revision cycle – Graphics which are created for other purposes have a revision cycle naturally different than the intelligence added to them. Preserving investments in metadata authoring requires careful planning and systems design.

By applying careful systems design concepts and incorporating a server-side graphics server, most, if not all, of the above challenges can be met and the cost of deploying graphics in Web applications can be significantly reduced.

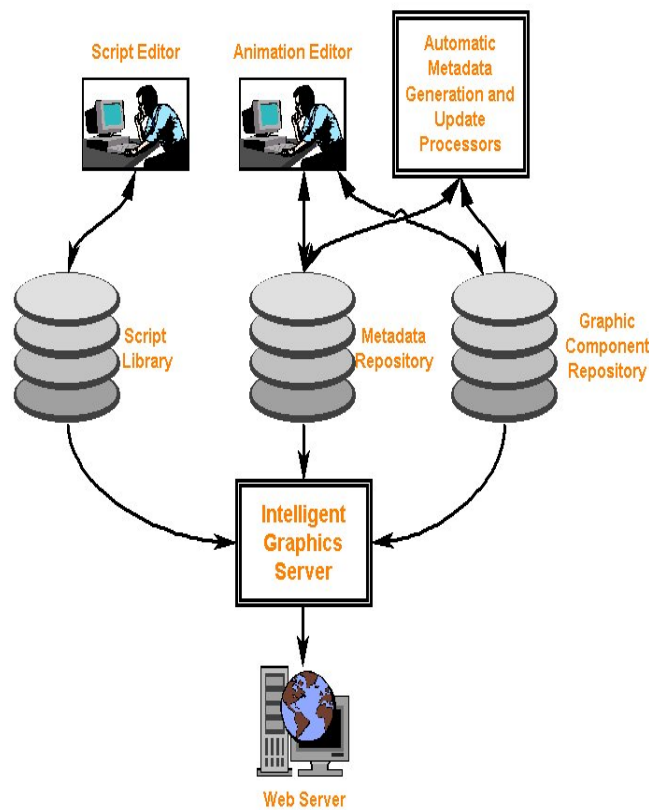


Figure 3. Component-based intelligent graphics publishing process

First, you must determine whether the graphics can be rendered on a Web page at sufficient quality without the user having to zoom in. The result of this analysis will help you choose the appropriate format. For

example, if you need to deploy complex illustrations such as contained in the illustrated parts catalog of a commercial aircraft, you will have to allow your user to zoom in or magnify the content, otherwise detail will be lost. One of the scalable vector formats such as WebCGM would be an appropriate choice for this type of graphic. On the other hand, if most of your graphics are color photographs which can be represented adequately at typical screen resolutions you should look first at GIF, JPEG, and PNG.

The next step in designing an intelligent graphics management system is to decide how you want your users to be able to interact with the graphics. The choices here range from publishing purely static raster graphics to fully animated, interactive, hotspotted, scalable graphics from reusable components.

The third step is closely related to the previous one. To provide intelligent graphics solutions which are easily maintained, it is very important to carefully analyze your graphics and break them into logical components. In some cases, these graphical objects will be bound to intelligent metadata such as hyperlinks. In other cases, the graphical objects will be used to create animations. In other cases, the objects will be created, when needed, from information contained in other databases. For example, the parts in Figure 1 can be associated with various other information via hyperlinks. If the user was repairing the equipment, the hyperlink could point to repair instructions. If a new part needed to be ordered, the hyperlink could point directly to the spare parts procurement application. If training was required, the hyperlink could point to a CBT (Computer-based training) module where the same graphical objects are used in an animation illustrating how to remove and replace the part.

Another example of intelligently building Web graphics from components is shown in Figure 4.



Figure 4. Static image created on demand from components

In this case, the background image of the product (in this case a computer system, can be composited with the retailer's logo, the current price, and the current status. The background image can be used over and over and the other information can be retrieved from the retailer's ERP system at the instant the Web page is requested.

If you choose your graphical objects carefully, you may be able to create them automatically from the source graphics or data. For example, if we had chosen to only use the callout numbers in Figure 1 as the hotspots,

the process of creating the hotspots could have been easily automated. By searching for the text strings in the drawing and performing a simple pattern match to filter out all text except callouts, intelligent objects can be created from the graphic.

Your ability to revise an intelligent graphic with minimal effort depends directly on how the graphics are maintained in the repository and how well the source graphics are structured. The best revision control strategy is to maintain a highly granular structure of graphical components and associated metadata, with version control on each small component. Then if one or more components need to be revised, new versions of the changed components can be added to the repository. An efficient, quality assurance process can then be designed that focuses attention on the changed components.

If the legacy graphics from which the intelligent graphics are derived are also structured into objects aligned with the structures maintained in the repository, then the process of incorporating a revised legacy graphic into the repository can be largely automated. The structures in the repository can be compared to the structures in the replacement graphic and changes can be detected. These changes can be "highlighted" to make the update process as efficient as possible.

If the legacy source graphics are not structured into objects, the existing graphics and the replacement graphics cannot be automatically compared, and the update process becomes more difficult. However, if the object structures are constrained to this which can automatically and reliably be derived from the source graphics, then the legacy graphics can be processed into objects as part of the revision process.

Summary and conclusions

Good graphics make Web applications easier to use and understand and provide very intuitive mechanisms for human users to interact with and control complex business processes and information. But until recently, creating and managing of large repositories of hand-crafted intelligent graphics was difficult and expensive.

However, by separating the graphics into reusable components or graphical objects, and by further separating the metadata associated with those objects, server-side intelligent graphics management systems can be deployed which combine the appropriate graphical objects and metadata into final-form intelligent graphics on demand.

Also, by utilizing one or more of the emerging scalable Web graphics standards such as WebCGM or SVG, these intelligent graphics can be served to Web clients very efficiently, with minimal bandwidth and maximum quality, thereby improving the users experience.

By maximizing the reuse of graphical components, minimizing the requirements for manually authoring intelligent behavior, automating the graphics revision process as much as possible, and utilizing the new Web graphics standards, the large scale deployment of intelligent graphics becomes economically attractive.

Bibliography

- [PNG] PNG (Portable Network Graphics) Specification Version 1.0 W3C Recommendation October 1,1996; <http://www.w3/TR/REC-png.html>
- [WebCGM] WebCGM Profile, Version 1.0; January, 1999; <http://www.w3/TR/REC-WebCGM.html>

- [GIF] GRAPHICS INTERCHANGE FORMAT(sm); CompuServe, Inc, 1989;
<http://ftp://ftp.ncsa.uiuc.edu/misc/file.formats/graphics.formats/gif89a.doc>
- [SVG] SWF File Format Specification; Macromedia, Inc;1998
- [WebCGM1] WebCGM - Industrial Strength Vector Graphics for the Web; CGM Open Consortium, Inc.;
Gebhardt and Henderson, 1/28/99.
- [SWF] W3C Working Draft: Scalable Vector Graphics; **[http://www.w3/TR/SVG](http://www.w3.org/TR/SVG)** .
- [JPEG] JPEG JFIF; **<http://www.w3.org/Graphics/JPEG>** .
- [ACGM] ActiveCGM Profile; Micrografx, Inc.; **<http://http:micrografx.com/intercap>** .
- [VML] Vector Markup Language (VML); May 13, 1998; **<http://www.w3.org/TR/NOTE-VML>**

Authors

John C. Gebhardt

Vice President, Product Architecture and Strategy
Micrografx, Inc.
Postal Address:
116 Defense Highway
21401 Annapolis
MD
USA
Telephone: 410-224-2926
Fax: 301-261-8358
E-mail: johng@micrografx.com
Web: www.micrografx.com

Dr. John C. Gebhardt — Dr. John C. Gebhardt, Vice President, Product Strategy and Architecture is a 30-year veteran in the computer graphics industry and has provided leadership in the development of electronic technical publishing products since the early 1980s.

Dr. Gebhardt is active in the development of international standards for the storage, interchange, and presentation of graphical information. He is an active member of Insight H3, the U.S. counterpart to ISO SC24. He was a coauthor of WebCGM and was an active participant in the development of VRML97, the new ISO standard for 3D graphics. He is a director and officer of CGM Open, an affiliate of OASIS which promotes the use of the CGM standard. Dr. Gebhardt also serves as an observer on the Graphics Task Group of the Air Transport Association (ATA).

Dr. Gebhardt has spoken on WebCGM and intelligent graphics at XML 98 in Chicago, XML 99e in Granada and at XML 99 in Philadelphia. He holds bachelor of science, master of science, and Ph.D. degrees in engineering from the University of Michigan.

Greg Gallant

Senior Systems Architect
Micrografx, Inc.
Postal Address:
116 Defense Highway
21401 Annapolis
MD
USA
Telephone: 410-224-2926
Fax: 301-261-8358
E-mail: gregg@micrografx.com
Web: www.micrografx.com

Greg Gallant — Greg Gallant is a Senior Systems Architect at Micrografx. He is currently leading the design of a middleware product which utilizes XML, WebCGM, and other W3C standards extensively. Mr. Gallant's professional interests include distributed systems and Internet technologies.