Abstract

The confluence of technical and business advances in the printing and publishing industry is discussed and used to frame the justification and purpose of developing an industry architecture. The components of an industry architecture (i.e., model, data dictionary, identification of standard components, and architectural constructs) are presented in detail. Options for systems-to-system integration technologies are compared, and the OMG's CORBA specification is singled out and then explained to a limited degree. The paper concludes with a two-step proposal for creating an industry architecture through GCA and OMG offices.

Part I —History, Foundation & Justification

Common Ground

Yesterday I got an interesting call from a market researcher working on an industrial equipment resale projection report. Printing equipment was one component of "industrial equipment," which is how he eventually found his way to me. He had already referenced many sources and had all the numbers, quantities, categories, and so forth you would normally expect, but he needed more. He wanted to gather opinions of where technology was going in terms of the "big picture" as well as what the "big picture" would mean to buyers and sellers of equipment and senior management. We discussed leading edge trends for awhile when he said something that got me thinking; "I've been talking to people from other industries and it almost seems that everyone is working on the same plan. It's almost eerie."

In the Spring Edition of *The Spectrum Letter*, I summarized Thomas Stewart's premise;[1]

> "According to Stewart, companies have learned to replace capital, in the form of material, equipment, and energy, with knowledge. Every product has a *knowledge content* — the knowledge required to make the product. As companies and industries acquire better knowledge of science, process control, engineering, and markets they learn to produce their products with less energy, less material waste, less labor, and so forth. However, in most cases, product prices do not erode. As a result, Stewart argues that consumers in effect are paying more for

the knowledge content of their purchases than they were twenty, ten, or even five years ago. As a result, knowledge has become a competitive weapon."

This trend is hardly surprising to anybody who is current with recent business news. What is surprising is that fundamental technologies, changes in business practices, industry trends across very different industries are starting to "line-up." There is so much in common among very disparate industries, it's almost eerie.

The Plan

The buzzwords, old and new, are numerous: supply chain management, digital asset management, process integration, channel management, e-commerce, just-in-time or zero inventory management... The technology candidates are even more numerous: XML, IDL, PDES, ICE, DCOM, EDI, CORBA, CIP3, WIDL, PDF, OPSLA, ICC, ... How could there be a plan when it's seemingly impossible just to get a grasp of what everything is? Let me skip ahead so that we can make sense of the particulars later.

The Goal

The goal is to sell. *Everything else must support this single objective in one form or another by bringing the product in line with the buyer's expectations, eliminating all "pauses" in the production process in order to create a fluid stream from material and ordering to delivery and payment, and by eliminating costs that are not essential or may even be impediments to the sale.*

This is my conclusion and the only point to remember if nothing else is to be remembered, (hereinafter referred to as the "Goal.") If you truly take this single statement to heart, everything else in this paper will seem *a priori* and seamless, and you will find it relatively easy to pick and choose your way through the throng of new technologies and product introductions that the market tosses at you every day.

The companies that work towards the Goal the best will become the market's leaders, but there is a hitch. Any company that uses the skills of multiple workers (thank you Mr. Adams) is dependent on cooperation for profit — cooperation between workers, between producer and supplier, between producer and distributor, *and* between seller and buyer. Competitive pressures have always pushed us towards improving cooperation but until the early to mid 1980's improved cooperation meant better management techniques, Deming, statistical process control, TQM and so forth. Jack Welch's, "goal of reducing defects to the point where errors would be almost non existent: 3.4 defects per million, or 6 sigma"[2] leaves one wondering how much more can be wrung out of your operations.

Naturally, businesses have turned to cooperative relationships with external business partners to find ways of streamlining operations in pursuit of the Goal, but then there is another "hitch" to contend with. To streamline cooperative operations (i.e., supply chains, channels, etc.) partners must come to an agreement on mechanics of cooperation. These "mechanics" include all the methods and means of communications we've all become intimately familiar with: common file formats (PDF, TIFF/IT, XML, etc.), common protocols (SNAP, GRACoL, SWOP, PROSE, etc.), and common connections (the Internet, WAM!NET, AP Adsend, etc.) We have learned several very important lessons in the process of establishing these very basic mechanics:

1. **Cooperation has many arms.** The "supply chain" within a cooperative is usually a much larger network than we initially expected. One example of a technology that is targeted towards streamlining the process and improving the "supply chain" is computer-to-plate (CTP) technology. A printer or prepress service who invests in CTP technology must not be concerned with only the platesetter and plates, but must also consider the RIP and front-end software, the file formats that are supported, the software and workflow of the customer, the platform the customer uses, the type of work the customer specializes in, and the needs of their customer's customers in terms of color, quality, time to market, and so forth. Following the "network" in another direction, the CTP investor must understand the imaging technology of the platesetter well enough to determine longevity of the laser, replacement cost of the laser, the exposure process and plate material requirements, and plate availability. Suddenly, the printer is concerned with computer technology, software design, systems integration, materials science, laser technology, and a host of other areas of expertise that are not normally attributed to printers.

2. **Open implies “very wide.”** Our definition of “open systems” changes as cooperative networks expand out of departments, beyond companies, and between market segments. Lets use color management technologies for our example here. At one point a prepress service, whose product was graphic arts film, could buy a system from one vendor with little regard for compatibility with other systems. As customers moved from mechanicals to digital art the prepress service then had to select systems that were compatible with desktop publishing systems (PostScript, TIFF, etc.) and the first generation of color management systems arrived on the scene. Then customers started bridging content to other media and re-using content in various forms of print. Now we are working to develop International Color Consortium (ICC) profile mechanisms that can be attached to almost any content type and smoothly communicate color definitions from digital cameras, to monitor viewing (design & Internet delivery), to print production.

3. **There’s no such thing as too flexible.** Since every print job is unique, arguing the merits of flexible manufacturing with the print community is “preaching to the choir.” However, we have learned that flexibility has a time component as well. Equipment and systems that are composed of smaller components that can readily be replaced or updated provide a competitive edge over the “big systems” of yesteryear. Top-down (or bottom-up) large systems integration projects that are either customized or proprietary become harder and harder to replace as they age. As hardware, software, and file formats become obsolete, the cost of replacing big systems grows and the big systems become a competitive disadvantage, or they spawn problems such as the Y2K bug. We have learned that it's more practical in the long run to design systems with components that can be replaced individually.

At the Research and Engineering (R&E) Council's *Smart Plants* conference in March of this year, Larry Bryant of Harper Collins said that, "The printing industry's movement into the digital world has been on a vertical path, creating many digital islands."[3] When Harper Collins reorganized their paper-back and hard cover book supply chain, they were able to reduce returns by 57%. During the same conference, Robert Potts of Mack Printing said that they are moving some jobs from digital file transfer to press in just 2 1/2 hours. Although Mack Printing is using standard such as SGML for file construction, they spend 35% of their time manually tracking transactions. Why? They need to integrate the business side of their operations in order to employ real-time job submissions and traffic management in such a way as to allow customers to "see" estimating, scheduling and workflow. We heard a similar story at last Fall's Spectrum '98 conference. Members of Jeff Bartol's (formerly with Colorhouse and now employed by Banta) digital asset management (DAM) panel said that customers want to be able to review and control assets that are managed in DAM systems at multiple vendor sites. However, there is no current standard method for querying these systems or constructing inter-enterprise integrated reporting systems. Jeff called upon GCA to work just such a set of standards, which brings me back to *the plan.*

As illustrated above, cooperative operations have already led the graphic arts to the point where we must integrate more closely with customer and supplier organizations that were once considered outside our "domain." Our customers are also facing this very same dilemma. For instance, automotive engineers now hand off illustrations and drawings to their marketing group to use in advertising — a process integration issue that nobody was thinking about ten years ago. Not surprisingly, cooperative relationships and candidate technologies are emerging, via natural selection (thank you Mr. Darwin) that meet certain fundamental principles:

The Principles of Cooperative Operations

1. Freedom of Assets. All assets must be free of encumbrances. In terms of materials and products the objective is to minimize all uses of materials that are not part of the Goal, to include: transfer storage points, time in inventory, waste and spoilage. In terms of knowledge assets (text, graphics, etc.), the data must be kept in a form that is not dependent on any one software application, storage device, or computer platform. All assets must be organized to move efficiently across distance, time, and location in support of the Goal.
2. Trust. Participants in cooperative operations must trust their partners ability to uphold their responsibilities within the cooperative ethically, fairly, and profitably. Those who do not trust the principles of cooperative operational relationships will not join in and will perish. Those who are not trustworthy will be asked to leave cooperatives and will also perish.
3. Faith. Participants in cooperatives must have faith that their pursuit of the Goal, although time consuming and more costly in the short-term, will ultimately lead to growth and profit. Or as Mike King of Amway Corporation said at the *Smart Plant* conference, you have to “give up, to get!”
4. Flexibility. All systems, physical and informational, must be flexible on the component level and must be adaptive to all future requirements.
5. Pursuit of knowledge. Each participant (or system) in a cooperative adds value through their knowledge of a particular aspect of the process, and each participant must improve its knowledge continuously, otherwise it, and possibly the cooperative, will become stagnant and will perish at the hands of those who learn to do their part of the process better.

In addition to changes in the way we do business, the advent of cooperative operations also has changed the way we approach technology. As each cooperative “network” spreads and reaches out to other industry segments, there are both an exchange of ideas, needs, and requirements and a growing consensus on common ground and principles. Not surprisingly, many industries are now in the process of developing methods of process integration such as in-line preventative diagnostics, knowledge or asset management systems, supply chain management, and so forth. Each industry has grown from its own "digital islands" and is now trying to bridge those islands. In the process, similar solutions are being invented "a new" all around the world. It almost looks as if there is a plan, but it's not so much a plan as it is a process.

- In order to make cooperatives possible, each group needs to commit its business practices to some form of electronic communications. (e.g., electronic data interchange (EDI), electronic content rights, syndication & security, & "e-commerce.")
- In order to achieve Freedom of Assets, each group needs to select file formats and conventions that are independent of software and hardware. (e.g., SGML, XML, PDF, TIFF & TIFF/IT, standard color characterization data sets (such as ANSI CGATS TR001) and ICC Color Profiles.)
- In order to achieve systems Flexibility, each group needs to select systems-to-systems command, query, and integration conventions. (e.g., CIP3, CORBA/OMA, WIDL, DCOM, JavaBeans, etc.)

Towards an Industry Architecture

Today we have the option to create an industry architecture for the printing and publishing industries. All of the building blocks are in place and the decision is simple. If we tackle an industry architecture cooperatively we can save ourselves millions of dollars and thousands of staff hours of wasted effort. Each time that barriers have fallen between entities someone has paid the price, and each time the price gets bigger. If your color prepress systems of the mid-1980s could not support customer DTP workflows, it was replaced. If your production workflows could not support re-purposing to the worldwide web in the mid-1990s, they were replaced. If your business cannot support completely digital workflows, it's on the way out. We are now seeing seamless integration with both customer and supplier systems becoming a requirement of business. If we don't cooperate as an industry in developing an industry architecture, we may lose revenue to other media types, or multiple "best practices" will develop that will eventually come in conflict with each other, and somebody will lose. If we cooperate as an industry, we stand to attract new sources of revenue.

There are two primary "classes" of information business and production. These two classes of information types have been kept apart, but in the last few years we have seen more and more companies and applications interweave these two classes of information. On one hand you may ask, what does the accounting system have to do with my imposition utility or preflight utility? However, as we become more sophisticated systems users we have begun to ask whether or not our workflow and scheduling systems can't be made to control the flow of job assignments. If so, why can't we get the workflow systems to feed into management information systems, along with accounting information, to provide management with more realistic financial projects? Or perhaps, to help the purchasing manager better coordinate purchases? These are two specific examples and there are numerous similar cases, but you get the idea.

At the industry level, our standards are moving in the same direction. There are EDI tools available that are used to buy paper, send shipping manifests, *and* provide TQM data back to the paper manufacturer. Adobe has provided a portable job ticket feature for PDF. SPACE X12 [4], approved for release this May, provides for the exchange of purchase information as well as production instructions. What is evolving are two new information types that include both business and production types of information: form and function.

Form. Form addresses our first principle of cooperation: Freedom of Assets. We are moving towards data types that free our assets in terms of time, distance, and location. In the case of final document assembled for print production the file formats at TIFF/IT and PDF/X. In the case of most other types of data, the industry is moving to XML. Although most readers will are now familiar with XML, let me say that XML is a variant of SGML in part simplified for optimal use over the Internet. (For more information, please see "What is XML" at www.gca.org) XML is providing exciting new features for publishers such as ICE, a standard method for syndication of information asset for Internet Content Exchange. Furthermore, the XML/EDI initiative is working on features necessary to move X12 constructs to XML. Although the data will remain the same in terms of data type, record lengths, and so forth (in order to guarantee portability from X12 to XML/EDI), the identification or wrapping of the data will allow you to use Internet tools to manage the data, while still providing for machine-to-machine data transfer. The GCA Board has already decided that all GCA EDI transactions will migrate to XML once the X12 standard provides a migration path.* We have made a lot of progress on "form" and what we'll be discussing from hereon out is "function."

Function. By "function" I mean the methods, standards, and data forms used to link business and systems together in order to make something happen; such as instructing a remote proofer to run a test chart, or asking an asset management system to search for certain information or return a data file in a format that can be used to form a progress report. This is an important distinction. The purpose of building an industry architecture is not to write file format standards all over again. In fact, it was not really possible for our industry to create an industry architecture until we had the "form" standards in place; they literally are the building blocks. An "industry architecture" will consists of:

1. A model of the basic process of publishing and printing which provides a framework.
2. An understanding of how each of the "building blocks" (e.g., PDF/X, PROSE, CIP3, etc.) fit into the model.
3. A set of common terms and definitions, or rather an industry data dictionary.
4. A set of constructs for creating and executing functions.

* Several technology experts supporting both XML and other candidate EDI technologies have stated that the embedded business rules and rigidity of X12 transactions are a hindrance to progress. However, as a developer of EDI transaction specifications, GCA has learned first-hand that the business community requires enforcement of legal and liability considerations in all business transactions. Executives will *not* support technology that does not guarantee their need for maintaining the integrity of contract documents, which is what business-to-business EDI is all about. Popular Internet based "e-commerce", such as on-line retail purchasing, is easy and simple. Some technology experts think that the ability to modify EDI structures on the fly, or for receivers to take what they want and discard other elements of an EDI transaction, is an advantage of new technologies. *This is an enormous mistake*. XML will change *how* EDI works, but will not change *what* EDI does. For instance, in the development of SPACE X12, advertising agencies required that there be provisions for embedding large legal statements in the form of text. For example, airlines require text that covers contingencies such as pulling advertising at the last minute in the event of a air plane accident. The obvious solution seemed to be to place a URL and page name where boiler plate legal clauses could be found if needed, rather than cluttering the transaction with static and repetitive text. However, for the airlines, who have been very progressive in other areas, this is a drop-dead issue. One of the primary motivators for moving EDI to an XML format is to bypass expensive private networks and dedicated connections in favor of the Internet; however, X12 transactions can be exchanged on the Internet today. Senders can simply write the transaction set to a file (X12 is ASCII based) and upload it to a receiver's FTP site. The receiver's can easily be set-up to act as a "hot folder," automatically forwarding files to an X12 translator for verification, database upload, and to generate the required response. The primary advantage to XML is that transactions can be reviewed by a user through a browser on both ends of the transaction.

Part II — The Components of an Industry Architecture

A Basic Model

The first step is to create a basic model for an industry architecture. The purpose of a model is "to capture and precisely state requirements and domain knowledge so that all stakeholders may understand and agree on them." [5] However, in information systems development a model can be used at various levels. For instance, some modeling tools are used to detail the full specifications of a final system and these tools may be used to generate programming codes. Models may also be used to document the thought process of systems analysis or to abstract key concepts of a system. In our case, the first modeling effort will be at the highest level: a complete view of the process components of print and electronic publishing. Since our goal is an industry architecture, we must take care not to create a model that is too specific and rigid. We must take into account that different business relationships use very different workflows; however, most graphic arts processes are the same from market segment to market segment.

There are two resources that are appropriate starting points for creating a basic model: The IPDC Model and AMPAC. GCA's Industry Policy and Direction Committee (IPDC) set out to reach an agreement over business issues in digital production. They too ran into the problem of multiple workflows and decided to build a generic model that they could use as frame of reference. Appendix A contains an illustrative version of the IPDC's model.[6]

1st Layer	Production process	Material	Design / Spec	Dictionary
2nd Layer	Classification of process	Name of material	Object	00
3rd Layer	Functional element	Functional classification	Element	**
4th Layer	Setting parameter	Characteristic parameter	Parameter instructed	**
5th Layer	Relation between the parameter and others			Possible values
6th Layer	Function for getting the value of parameter (pointer)			Interpretation of values (pointer)

Figure 1 -- AMPAC Layers

The second starting point is the Architecture Model and Control Parameter Coding for Graphic Arts or "AMPAC."[7] AMPAC is an ISO Technical Committee 130, Working Group 2 working draft that is being developed in cooperation with the Japanese Imaging Processing Technology Standard. AMPAC seeks to standardize encoding of production parameters from design, plate making, printing and so forth so that these parameters can be more easily exchanged between all nodes in the production process. AMPAC incorporates a six-layer (see Figure 1) hierarchical encoding system that tags each production process and process parameter with an alphanumeric ASCII data identifier. Each identifier inherits a prefix from it's parent layer. For instance, in Figure 2 you will notice that all level 2 materials (e.g., paper, film, presensitized plates, blankets, ink & fountain solution) begin with "12," which is the identifier of the materials in level 1. Each subsequent level becomes more detailed. For instance, the third level of paper encoding includes properties of paper such as geometrical properties, strength and so forth. The fourth level of AMPAC encoding for the geometric properties of paper include data items such a roll width and caliper.

1st layer	2nd layer
[00] Specification of parameter values *1	[0000]
[02] Design	[0202] outline of products
	[0204] design concept of document
	[0206] page layout
	[0208] comprehensive
	[020A] object(page element)
	[020C] change in object
	[020E] instruction(keyline)
	[0210] selection paper stock
	[0212] instruction for bookbinding
[04] Text processing	[0402] input composing basic layout
	[0404] basic style
	[0406] special style
	[0408] ruby
	[040A] heading
	[040C] cutting note
	[040E] separation
	[0410] commentary
	[0412] table
	[0414] rule
[06] Text processing device	(Under construction)
[08] Image input	[0802] contents
	[0804] schedule
	[0806] preparation of input
	[0808] image input work
	[080A] preparation of line art image input
	[080C] line art image input work
[0A] Image processing	[0A02] disk set-up
	[0A04] platemaking instruction
	[0A06] edit
	[0A08] save
[0C] Image assembly	[0C02] line art image tint making
	[0C04] line art logo input
	[0C06] image combine
	[0C08] line art image combine
	[0C0A] trapping
	[0C0C] registration
	[0C0E] layout sheet
	[0C10] proof

1st layer	2nd layer
[0E] Output processing	[0E02] output preparation
	[0E04] film output
	[0E06] press plate output
	[0E08] proof
[10] Image processing device	(Under construction)
[12] Material	[1202] paper
	[1204] film
	[1206] presensitized plate
	[1208] blanket
	[120A] ink
	[120C] fountain solution
[14] Web-fed offset press	[1402] paper feed
	[1404] printing
	[1406] dampening
	[1408] image transfer
	[140A] drying
	[140C] cooling web bath
	[140E] folding device
	[1410] stacker bundler
[16] Sheet-fed offset press	[1602] paper feeder
	[1604] press
	[1606] delivery
	[1608] management
	[160A] quality check
	[160C] safety
[18] Gravure printing press	(Under construction)
[1A] Letterpress	(Under construction)
[1C] Screen printing machine	(Under construction)
[1E] Non-impact printing	(Under construction)
[30] Bookbinding	[3002] saddle stitcher
	[3004] adhesive side stitcher
[40] Converting	[4002] flexible packaging laminate
	[4004] flexible packaging extrusion laminate
	[4006] flexible packaging slitter
	[4008] box forming machine
	[400A] carton diecutting and creasing
	[400C] sack
[90] Process evaluation	[9002] image quality
	[9004] image quality control
	[9006] operating controller

Figure 2 -- Encoding Levels 1 & 2.

What AMPAC and the IPDC model provide are a head start: within these two sources most processes are identified and organized in a flexible yet thorough manner. AMPAC may act as a catalog of process components and the IPDC model provides a framework onto which these process components may be placed.

An industry model should also be built in symbolic language that itself is platform independent and portable to many applications. The Unified Modeling Language (UML) is a graphical modeling language that uses four kinds of graphical constructs: icons, two dimensional symbols, paths and strings. Before UML there were over 60 available object-oriented design and analysis methods on the market. UML is a standard conglomeration of modeling methods that can be used to create static structure diagrams, use case diagrams, sequence diagrams, collaboration diagrams, state chart diagrams, activity diagrams, and implementation diagrams. UML is endorsed by the Object Management Group and ISO, and is jointly owned by a slew of leading technology companies.[8]

According to Alan Pope, "analysis of the business model produces an architecture (high-level design that is free of implementation detail. It is the model after its first transformation into Von Neumann space. In other words, it moves the problem space from the real world into that of computer constructions. From the business model, commonalties are further abstracted; identification occurs for entities and their operations."[9] Other factors or "commonalties" in the model may include organizational relationships, interactions over time, and practical considerations and constraints. According to Pope, as the model builders define *how* entities relate to each other, as opposed to just defining *what* the entities in a model are, business requirements are transformed into an architecture model.

AMPAC does not describe relationships and the IPDC model does not identify sufficient detail of *what* all the process entities are, nor is it described in a sufficiently rigorous and functional symbology. However, by building upon AMPAC and the IPDC model within a formal symbology such as UML, we could hope to build a fairly robust industry model relatively quickly.

Gathering the Building Blocks

The second step in building an industry architecture is to identify the "building blocks" of our model. Many of the business and technical rules of the printing and publishing industries already exist and are contained in industry standards and specifications. In addition to knowing the basic processes and relationships of business entities within our model, we need to know the rules they must adhere to and the language that they have established. These can be gathered by coming to agreement on the standards to be incorporated into the model and then by examining each of these "building blocks" and determining where in the model they fit. The following table is an example of the standards and specifications (left-hand column) that we may select and the business entities to which they may be attributed (right-hand column.)

	Planning	Authoring /Creative	Prepress & Pre-Publishing	DAM	Archives & Syndi-cation	Press	Bindery & Finishing	Postal & Print Delivery	On-line & WWW Delivery
AMPAC	X	X	X	X	X	X	X		
CIP3			X			X	X		
DCS			X		X				
EDI	X		X	X	X	X	X	X	X
EMS		X	X						
GIF		X	X	X	X				X
ICC		X	X	X	X	X			
ICE		X	X	X	X				X
IPDC	X	X	X	X	X	X	X	X	X
JIFFI	X	X	X			X	X	X	
OpCodes†						X			
PDF		X	X	X	X				X
PDF/X			X	X		X			
PJTF	X		X			X	X		
SNAP, GRACoL & SWOP	X		X	X		X	X *(Gracol only)*		
TIFF & JEPG		X	X	X	X				X
TIFF/IT			X	X					
XML		X	X		X				X

† GCA standard operations codes for web press operations are nearly 25 years old and are commonly used by most web press management and information collection systems. These codes are so old and so common they are often over looked and often user don't know where they originated from.

Table Glossary

AMPAC	Architecture Model and Control Parameter Coding for Graphic Arts.
CIP3	*International Cooperation for Integration of Prepress, Press, and Postpress*, Fraunhofer Institute for Computer Graphics, Darmstadt, Germany. ftp.igd.fhg.de/outgoing/igd_a1/cip3.pdf
DCS	Desktop Color Separations file format, Quark, Inc., Denver, Colorado.
EDI	Electronic Data Interchange. In this table EDI is used as shorthand notation for all graphic arts EDI transaction types to include SPACE/X12, PROSE, Mail.dat, Ship.dat, EMBARC, SupParMan, and so forth, which are managed by Graphic Communications Association, Alexandria, VA.
EMS	Electronic Mechanicals Specification, GCA Standard 118, Graphic Communications Association, Alexandria, VA. 1993.
GIF	Graphic Interchange Format, Unisys Corp.
ICC	International Color Consortium, Specification ICC.1: 1998-09, File Format for Color Profiles.
ICE	Information and Content Exchange protocol, Worldwide Web Consortium
IPDC	GCA's Industry Policy & Direction Committee (IPDC) created *The Business of Digital Production*, which cover the responsibilities and liabilities of each participant in the process.
JIFFI	Job Instruction File Format for Industry, PGC/GCA Standard 138, Version 99-1.0, Graphic Communications Association, Alexandria, VA. 1999.
PDF	Portable Document Format, Adobe Corporation, San Jose, CA.
PDF/X	PDF for Graphic Arts Interchange, American National Standard, CGATS.12, NPES, Reston, VA
PJTF	PDF Job Ticket Format, Adobe Corporation, San Jose, CA.
SNAP	Specifications for Non-heatset Advertising Printing, Printing Industries of America, Alex., VA
GRACoL	General Requirements for Applications of Commercial offset Lithography, version 3.0, GCA, Alexandria, VA 1999.
SWOP	Specifications for Web Offset Publications, 8th Edition, SWOP Inc., New York, NY. 1997
TIFF	Tagged Image File Format, Adobe Corp., San Jose, CA.
JEPG	ISO Joint Photographic Experts Group file format.
TIFF/IT	ISO12639, Graphic Technology — Prepress digital data exchange — TIFF/IT
XML	Extensible Markup Language (XML), 1.0, Worldwide Web Consortium Recommendation 10 February 1998, http://www.w3.org/TR/REC-xml

Creating a Common Language

The third step in building an industry architecture is to create a language from our model, or rather an industry data dictionary. This is also the point at which we begin to migrate from a simple model towards the architecture itself.

Each of the "building blocks" in our model use terms and data definitions that often overlap. In appendix B you will find a couple of pages from JIFFI that include a "schema" and a "data field discussion." The data field discussion contains an item called "Stock," which in AMPAC (above) is called "Paper," or rather "1202." This conflict of terms is a great example of the need for an industry data dictionary. An industry data dictionary would use a format similar to the JIFFI illustration, but would display like terms and definitions side by side, and it would add a generic or “standard” identifier. Each of the above reference “building blocks” would be analyzed and all data definitions that pertain to our industry model would be extracted and added to the industry data dictionary. This would provide managers and developers with a useful lookup table, and would provide generic identifiers that can be used to create common language between systems.

The Architecture — Creating and Executing Functions

The final step in building an industry architecture is to build constructs that allow systems to address external systems, make remote procedure calls, and generate a response from the remote system. At this point we have all the basic elements that we need: a model that describes entities and their relationships to one another (e.g., “objects”), an identified set of rules and formats, and a list of generic terms and definitions that can be shared between systems. To create architectural constructs we need to do two more things: first, select a systems-to-systems integration mechanism, and second, select a procedure for prioritizing and defining construct requirements.

Selecting a mechanism

Although there are widely used standards, such as the Structured Query Language, that allow users to query databases and other data management tools available, there are only a handful of systems-to-systems mechanisms to choose from: DCOM/COM, JavaBeans, WIDL, and CORBA. In selecting a technology to apply within the printing and publishing industry we can narrow the field quickly with a few criteria. First of all, our choice must support both Windows and Apple platforms, as well as server operating systems such as NT and UNIX. The printing and publishing market is fairly unique in that some segments, such as design and layout, are predominantly Apple based.

Our choice must also be fairly robust from a technical point of view. It should have proven security, concurrency (dealing with multiple simultaneous sessions), etc. features and it must be compatible with legacy systems and multiple data types. Furthermore, our choice should offer lots of choices in terms of off-the-shelf tools and software applications. The following table provides a simple comparison.

Mechanism	Advantages	Disadvantages
Distributed Common Object Model (DCOM) /Component Object Model	• Supported by Microsoft • Relatively simple • Directly supported in MS Windows operating systems • Short development cycle	• Limited to some Microsoft operating systems • Not portable • No open process for standardization, vendor controlled specification • Limited features • Not a standard
JavaBeans	• Platform independent • Supported by Internet applications • "Write once, use many" portability	• No open process for standardization, vendor controlled specification • Few applications and tools • Not a standard
Web Interface Definition Language	• Platform independent • Based on XML syntax • Internet friendly • Portable • "Human readable" syntax	• Very new and unproven • Limited systems-oriented and programmer oriented features • Few applications and tools
Common Object Request Broker Architecture	• Directly supported by UML tools • Ten year history of implementation • Hundreds of supporting companies and applications • Very mature and robust programming features • ISO standard • Open standards development process • Portable to all systems, including legacy systems • Mapping to C, C++, Smalltalk, OpenDoc, Java, COM, etc.	• Often criticized and "complex" • Longer development cycle • Requires a greater degree of training and experience on behalf of programmers

DCOM/COM does not meet our platform criteria. WIDL and JavaBeans are promising, but very new and still developing. Goldfarb and Prescod say of WIDL, "It provides 80% of the capability of conventional IDL [interface definition languages], with only 20% of the complexity."[10] Unfortunately the missing 20% of capability includes features such as security and concurrency that are required in mission-critical systems. According to Sun Microsystems, "JavaBeans is a portable, platform-independent component model written in the Java programming language. JavaBeans works with any network model (i.e., to communicate between components across the network), including CORBA, DCOM, etc. JavaBeans integrates well with CORBA IDL, which is an excellent solution for customers in a heterogeneous distributed computing environment with platform independent components."[11] However, JavaBeans is new and Sun only recently introduced a review system for JavaBeans.

JavaBeans, WIDL, DCOM/COM and other systems-to-systems mechanisms all emulate the Object Management Group's (OMG) Common Object Request Broker Architecture (CORBA). Only CORBA is developed and managed in an open environment by a not-for-profit organization. CORBA is far and away the most mature technology in this class.

How does CORBA work?

CORBA is really a suite of specifications and standards of which the ORB is only one component; but before we get into all that we should establish a basic understanding of how CORBA or any other "object-oriented" systems-to-systems mechanism does.

In the graphic communications industry we are familiar with plug-ins and extensions. For instance some of the best features available with products such as XPress or Photoshop are not provided by Adobe or Quark, Inc., but by third-party developers. These extensions or plug-ins are possible because the lead companies make available Application Program Interface (API) definitions to the third-party developers. In short, a software API describes individual functions that the main program performs along with the input that is required by the function, the command language that activates the function, and a description of the output the function will generate. This is the core concept of object-oriented programming: each object consists of the data, the software, and the rules by which they function. The rules may include systems rules such as data typing or memory management, or business rules such as establishing access privileges or authorization. If you know this information and have access the software and data you do not have to address the software at the code level, you can simply address the "object." Most software developers now develop code in "object" units that can be quickly gathered, modified, and reused.

However, not all API's are created equal, in fact, most API's are expressed in a language or "syntax" that is unique to the software developer. The powerful idea behind CORBA is to create a common method to negotiate syntax between two or more systems and to "broker" object requests between systems. Systems do not have to have an "API" with a CORBA interface either, even legacy systems can be addressed through CORBA's Object Gateway feature by wrapping (or "encapsulating") the legacy system in an external "object." "The wrapper serves as an interoperability bridge between a legacy system and software architecture [CORBA]. On one side of the bridge, the wrapper communicates using the legacy system's existing communications facilities. On the other side of the bridge, the wrapper presents other applications a clean interface that provides abstract services."[12] The CORBA specification even provides for creating "bridges" between CORBA and other ORB-based methods such a COM and JavaBeans.

CORBA Features

Keahey provides an excellent short introduction to CORBA.[13] In Figure 3 there is a "client" (the object requester) and an "implementation" (the target application.) The client can communicate through the "ORB Core" through a variety of mechanisms. There are several types of ORB cores available to CORBA users. On option is the Internet Inter-ORB Protocol (IIOP) and another is the General Inter-ORB Protocol (GIOP) these two protocols use the popular IP or "Internet Protocol" and provide the communications pathway for the exchange of objects under CORBA. The basic tool kit focuses on the use of "stubs" and "Skeletons" written in OMG's interface definition language. IDL is the key component of CORBA and it is recognized by ISO as an international standard. An example can be found in appendix C, but don't be put off by the example: users need never to get *that* familiar with CORBA. In essence, the stub-skeleton connection requires that the client system knows what to ask the application for and how to make the request.

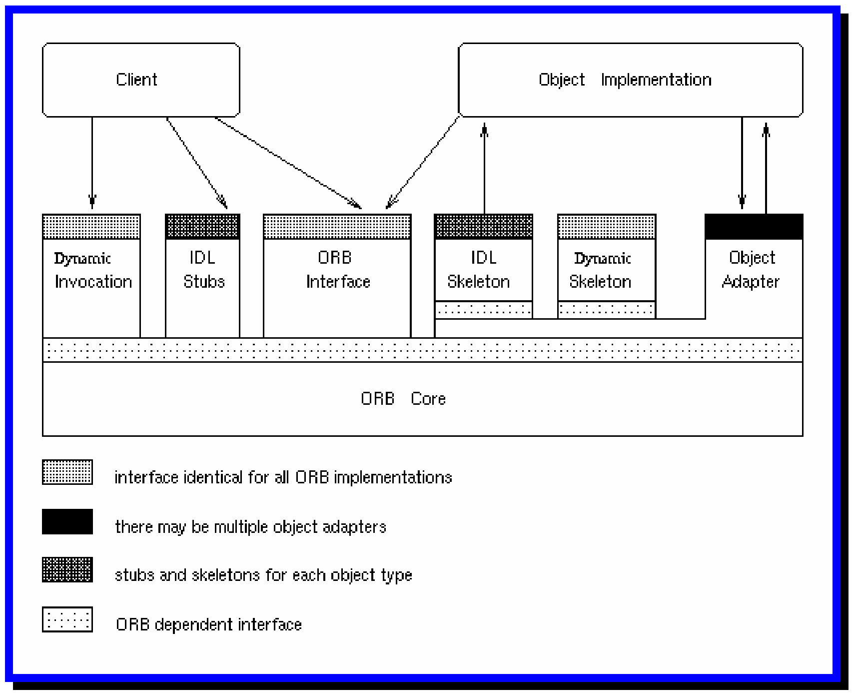

Figure 3

A CORBA feature that is even more pertinent to our industry architecture project is the "dynamic invocation interface" or "DII." The DII "is a generic facility for invoking any operation with a runtime-defined parameter list. A runtime interface description of the operations signature can be retrieved on-line from the CORBA Interface Repository. Using the metadata, a legal request can be constructed to a previously unknown operation and unknown object type."[14] The OMG has recognized that industry groups have unique requirements, in the areas of terms, definitions, and rules, and have introduced a process by which industry "Domain Task Force" groups can extend the CORBA features by creating industry-specific implementations that make use of features such as DII — This is where the printing and publishing industry can make use of what we have outlined in steps 1-3 above in order to define our unique CORBA requirements.

The OMG's suite of specifications includes many CORBA services that can be of use to the printing and publishing industry.[15] The availability of these services is proof of the robust and mature nature of CORBA.

- Naming service
- Lifecycle service
- Event notification service
- Persistent object service
- Relationship service
- Externalization service
- Concurrency service
- Object query service
- Properties service
- Licensing service
- Time service
- Distributed document component facility
- Object trader service
- Object collections service
- Systems management facility
- Object security service
- Transaction service
- Printing facility
- Person identification service
- Objects-by-value specification
- Notification service
- Mobile agents service
- Lexicon query service
- Currency service
- Workflow specification
- Firewall specification
- XMI specification (portability to XML)

A procedure for prioritizing and defining construct requirements

So where do we start? The first step is to develop the model and industry data dictionary as described previously. This can be done under the auspices of a GCA committee, with participation by OMG experts and in cooperation with key graphic arts organizations. In the lexicon of OMG this activity is the function of a “special industry group.” The data dictionary will function as our library of data types and terms that can be used as the foundation with which we can build printing and publishing CORBA facilities.

The second step is to form a Domain Task Force Group under OMG. GCA and OMG have already formed a partnership so that we can work together towards this objective. The Domain Task Force, operating under the procedures of OMG, defines requirements for CORBA solutions and writes a request for proposal. At this stage we would use the model we created previously to identify a limited and defined area to focus on. For instance, the group may choose to develop and RFP for pressroom automation and SPC data collection; or it may choose to focus on customer system-generated addressing to digital asset management systems; or perhaps a common reporting, scheduling and query mechanism for workflow systems used in all stages of production. GCA member participation by production managers and executives is necessary up to this point. Once the RFP is generated and approved via electronic ballot, it is released to the public via the Internet and technology companies respond with proposals. OMG has a detailed set of rules for this process, but the point to remember is that the CORBA experts return the *solution* and the users define the *need*. OMG also has an architectural review board that will help by validating that proposals are consistent with other CORBA features and services.

If you want to join the Printing and Publishing Industry Architecture project, please contact Jim Harvey at jharvey@gca.org or via phone at (703)519-8166. GCA will also provide members who sign-up for the project a copy of The Object Technology Casebook, by Harmon & Morrissey (Wiley 1996) which is a good management-oriented introduction to the subject, (*while quantities last*.) I also suggest that you obtain a copy (free) of *Object Oriented Business Modeling Requirements Analysis and Legacy Systems, by Alan Pope* from the Internet (see references below) and review introductory material and papers available at http://www.omg.org

Conclusion

There is ample need within the printing and publishing industry to improve our production process. Now that the basic file format standards, EDI specifications, and digital production specifications are settling into place, we have the opportunity to take the next step necessary to garner the promised benefits of digital production. We need to create an industry architecture that consists of four major components:

1. A high-level business and process industry model
2. An understanding of how each of the "building blocks" (e.g., PDF/X, PROSE, CIP3, etc.) fit into the model.
3. An industry data dictionary.
4. A printing and publishing industry architecture built in the CORBA framework.

The fourth component will provide users with a flexible, cost-efficient, and adaptable systems-to-system integration mechanism. Some of the benefits of using CORBA include:[16]

- The OMG IDL provides an operating system and programming language independent interface
- Due to this higher level of abstraction, the user does not have to be concerned with the lower layer protocols
- The user does not have to be concerned with the server location or activation state
- The user does not have to be concerned with the server host hardware or operating system
- Integration issues are simplified; for example, no longer does a client have to be concerned with byte ordering of data when transferring data between different platforms.

References & Bibliography:

[1] *A Practical Approach to Intellectual Asset Valuation and Accounting*, by James E. Harvey, *The Spectrum Letter*, Graphic Communications Association, Alexandria, VA. Spring 1998.

[2] *Revealed at last: the secret of Jack Welch's success*, by Michelle Conlin, Forbes, January 26, 1998.

[3] *R&E Council Smart Plants* Conference, March 16-17, 1998, unpublished notes by Jim Harvey, GCA

[4] *Specifications for Agency Publisher Communications Exchange X12*, Graphic Communications Association, Alexandria, VA 1999.

[5] *The Unified Modeling Language Reference Model*, by Jame Rumbaugh, Ivar Jacobson, and Grady Booch, Addison Wesly Longman, Inc., Reading, Massachusetts, 1999.

[6] *The Business of Digital Production*, James E. Harvey, editor, Graphic Communications Association Industry Policy & Direction Committee, Alaexandria, VA. 1999.

[7] *Architecture Model and Control Parameter Coding for the Graphic Arts (AMPAC)*, by Hiromichi Mishina and Larry Warter, pgs. 15-26, The 1998 TAGA Proceedings, Technical Association of the Graphic Arts, Rochester, NY, 1998.

[8] *Unified Modeling Language Notation Guide,* Version 1.1, Rational Software, Microsoft, Hewlett-Packard, Oracle, Sterling Software, MCI Systemhouse, Unisys, ICON Computing, IntelliCorp, i-Logix, IBM, ObjecTime, Platinum Technology, Ptech, Taskon, Reich Technologies, and Softeam, 1 September 1997. http://www.rational.com/uml

[9] *Object Oriented Business Modeling Requirements Analysis and Legacy Systems*, version 1.3, by Alan Pope, November 21, 1997. http://www.qds.com/oo_ad/ [Note: I strongly recommend this article for further reading. This "book" is detailed, yet it is easy for non-programmers to read.]

[10] The XML Handbook, Charles F. Goldfarb and Paul Prescod, Prentice Hall PTR, Upper Saddle River, NJ, 1998.

[11] JavaBean FAQ, http://splash.javasoft.com/beans/faq/faq.general.html#Q1, 28 April 1999.

[12] The Essential CORBA, Systems Integration Using Distributed Objects, by Thomas J. Mowbray and Ron Zahavi, John Wiley & Sons, Inc., New York, NY 1995.

[13] A Breif Tutorial on CORBA, Kathy Keahey, http://www.cs.indiana.edu/hyplan/kksiazek/tuto.html, 26 April 1999.

[14] Mowbray and Zahavi, 1995

[15] OMG TC Work in Progress, Object Management Group, http://www.omg.org/techproce...logy_Adoptions.html#tbl_MOF_Specifications, 26 April 99.

[16] Adapted from: Mowbray and Zahavi, 1995