

## BIOGRAPHICAL INFORMATION

Karen Bachmeyer  
Executive Manager, Research & Development  
Intergraph  
Mapping and Geospatial Solutions

### Specific Responsibilities

Ms. Bachmeyer joined Intergraph in 1986. Ms. Bachmeyer is responsible for many aspects of the development of Intergraph's G/Technology products. Specifically, she is the project manager for new software releases. Her activities include resource scheduling and tracking, requirements analysis, design reviews and direct management of several development teams. She is an active participant in the deployment of many G/Technology projects offering technical advice and ensuring timely delivery of software to customers. She is also involved in test planning and tracking for verification testing as well as new product release presentations.

### Past Experience

Ms. Bachmeyer began her career at Intergraph as a software analyst. Her focus in the development organization began in the metadata and configuration systems and has continued in this vein throughout her career. Ms. Bachmeyer has also held management roles within both the utilities and mapping organizations of Intergraph. Ms. Bachmeyer was the initial technical director for the Team GeoMedia partner program.

Prior to joining Intergraph, Ms. Bachmeyer worked as an analyst for Grumman Data Systems. She also taught computer science classes at the University of Alabama in Huntsville.

### Educational Information

M.S. – Computer Science, University of Alabama in Huntsville  
B.S. – Home Economics, Incarnate Word College, San Antonio, Texas

### Professional Memberships

GITA  
Institute of Electrical and Electronics Engineers

## WHAT DOES COTS REALLY MEAN?

Karen Bachmeyer

[Karen.Bachmeyer@Intergraph.com](mailto:Karen.Bachmeyer@Intergraph.com)

Intergraph, Mapping and Geospatial Solutions

Mailstop 23B, P.O. Box 6695

Huntsville, AL 35824-0695

### ABSTRACT

Commercial Off-the-shelf technology (COTS) is widely touted in many industries including GIS. This paper will first explore the term itself and how it typically manifests itself in GIS software. Discussion will include research that indicates the growing acceptance of COTS and the evolution of this approach in software development. Additionally, common mechanisms for choosing COTS software will be compared. With a basic understanding of the terminology, the paper will then discuss the benefits and pitfalls of COTS-based systems. This analysis will address many questions that the typical GIS user might have as they consider COTS:

Is COTS simply vendor hype?

What can my project expect in terms of real benefits?

What is my return on investment?

Does COTS mean “one size fits all” in my GIS?

Finally, the paper will analyze metadata integration within COTS software components. Metadata has two principle flavors – data description and application behavior. The GIS user will receive a clear understanding of the differentiation of the two types. He will also understand the varying degrees of flexibility to expect in order for metadata to guide and control an application. Examples of how metadata can change the face of the COTS software will be presented.

### DEFINING COTS

#### Defining the Term

While we can quite easily define the acronym –

- C – commercial
- O – off
- T – the
- S – shelf,

it is a more significant challenge to relate that to the operation of your business. Every industry is faced with the “build or buy” dilemma. The homemaker can build the loaf of bread from raw ingredients – or she can buy it. The GIS manager within a utility company can build a software application to map and analyze his network – or he can buy it.

In the strictest definition (Basili, 2001), COTS systems have the following characteristics: the buyer has no access to source code; the vendor controls its development; and the software has a nontrivial installed base. A more relaxed definition encompasses the combination of any set of software components or applications purchased and combined into a large software system. Almost without exception, every GIS-related endeavor will utilize a significant percentage of COTS software components. Even vendors of GIS software utilize COTS software components. The vendor utilizes micro components – smaller building blocks of specific functionality that enable him to build a more complex product. The utility company utilizes macro components –

whole software systems from multiple vendors that are combined to enable the utility to be more cost effective and to provide better service to his customer. Let's visualize COTS components as a ladder where the lower rungs are rudimentary software controls (e.g. a pre-built data grid) and the top rung is a fully integrated application that addresses all aspects of your business. As a GIS manager, where do you want to begin the ascent to complete integration? What are the costs associated with the various rungs?

### Manifestations of COTS in GIS

In most organizations, management of geographic data is a means to an end. The analytical use of geographic data enables better decision making by the organization. Better decisions ultimately lead to improved productivity, cost efficiencies and better customer service. As a GIS manager, the COTS components you choose and where you step on the ladder can be a predictor of how successful your organization will be in meeting its objectives.

If you start at a lower rung of the ladder, you have the luxury of greater control over your process and interface points. However, the development and maintenance costs of the software may far outweigh the benefits of control. If you start at a higher rung, the COTS software may force you to make radical changes in your business process. These changes may be just as costly to implement within your organization and may be accompanied by poor user acceptance.

GIS-related COTS products can be found at all rungs of the ladder. On the lower rungs, you might simply purchase data or a tool that renders the graphic image of the spatial data. You would invent the software to manipulate and manage that data. On the higher rungs, the software would be complete with workflow controls, data model and reports. You would be left to adapt your business process to using those workflows and leveraging its reports to make appropriate business decisions.

It's interesting to note that more than 99% of computer instructions come from COTS products (Basili, 2001). It's inevitable that your GIS activity will involve COTS software. Your project team simply cannot write every component needed to make effective GIS-related decisions and analyses. But, your evaluation of COTS is important so as to minimize risks and maximize benefits within the scope of your projects.

### ACCEPTANCE AND EVOLUTION

The world is changing and no organization can afford to write and maintain all the software it needs to be competitive in their respective industry. The Carnegie Mellon Software Engineering Institute chartered a group whose focus was to "learn, mature, and transition principles, methods, and techniques for creating systems from COTS products". This group has since evolved and expanded to address the improved creation and sustainment of systems comprised from off-the-shelf components – including non-commercial products. The activities of this group emphasize the challenges of using COTS products and offer advice for updating legacy systems to take advantage of COTS components. Of particular interest, we note that the traditional use of the waterfall approach for system development (system context & requirements > architecture & design > implementation) is supplanted by simultaneous consideration of the system context, the capabilities of current market products and viable architectures and designs (CM-SEI, 2004). That is, rather than designing your GIS system strictly from your requirements, you must look at

the available tools and determine how best they can be combined to meet your business process needs. This evaluation is iterative and on-going throughout the lifetime of your project and business.

Perhaps it's inappropriate to state that COTS has gained acceptance in industry. More precisely, we can say that COTS is both ubiquitous and inevitable. The specification of a Component Object Model (COM) as a programming paradigm first appeared in 1995. This approach to developing components for use in integrating custom applications could be considered the early foundation for COTS. In 1997, the Object Management Group (OMG) appeared as a not-for-profit consortium that produces and maintains computer industry specifications for interoperable enterprise applications. CM-SEI's first international conference on COTS-based software systems was held in 2002. This significant industry and academic attention to COTS and systems integration confirms that COTS-based software will play a role in most companies' software applications. It also confirms that the choices and challenges associated with deploying a COTS-based system are non-trivial.

### Mechanisms for Choosing COTS

One approach to defining the evaluation criteria is offered by Torchiano (Torchiano, 2002). This team presents a set of attributes to characterize COTS software and aid in a selection process. By analyzing your candidate products in light of these factors with appropriate weighting, you can better measure the likelihood of success for any of the products within your organization.

<b>Attribute</b>	<b>Description</b>
Product maturity	Years on the market and features stability
Market share	Ratio of the market covered by this product
Performance	Scalability – number of users it can scale to without sensible decrease in response time
Safety/Security	Capabilities offered for secure or safety-critical systems
Reliability	Fault tolerance of product
Hardware requirements	Characteristics required to run the product effectively
Product support	What facilities are available to support software development with the product
Documentation	What kind (web, on-line) and how substantial is the documentation
Usability	Degree of satisfaction from users
Learnability	How much time does it take to learn
Modifiability	How easy is it to modify – nonmodifiable, parameterized or API extensions; Are modifications supported where you need them?
Change frequency	How many releases per year
License type	How is the product licensed
Cost of use	Product cost (integration, deployment, administration)
Software requirements	Platform dependencies
Conformance	Standards that your organization requires and what standards the product conforms to as well as how it integrates with other COTS
Domain specific	What technology domain is the product applicable to

**Table 1 – Attribute Classifications**

You must define your own weighting criteria based on your organization's directives. Some attributes may be of little concern to your organization and can be omitted entirely. For example, the license type may be of no significance unless it has an implication on network access during product operation. Conformance to certain standards may be a gating factor in the evaluation.

For example, if the product must adhere to the FGDC<sup>1</sup> standard for metadata and it does not, then the product is not considered – no matter how well it matches other aspects of your organization’s needs. However, once a product meets your conformance requirements, then that attribute has no weight in the final selection.

Another approach to COTS evaluation is presented by Oberndorf (Oberndorf, 2002). The emphasis is on the process of evaluating products. The PECA phases are outlined here.

<b>Element</b>	<b>Description</b>
<b>Planning the Evaluation</b>	Forming the team; Creating a charter (scope and constraints); Identifying stakeholders; Picking the approach (depth of evaluation, strategy for selection, number of iterations of selection to filter candidate products); Estimating resources & schedules
<b>Establishing Evaluation Criteria</b>	Identifying evaluation requirements (system requirements as well as additional expectations and corporate requirements); Constructing criteria (capability statement and a quantification method)
<b>Collecting Data</b>	Execute evaluation plan against the products Techniques for collection – literature review, vendor appraisals, hands-on experiments
<b>Analyzing Results</b>	Data consolidation; Data analysis (reasoning about consolidated data to enable a decision) Making recommendation

**Table 2 – PECA Process**

This process is not always sequential. That is, as you begin establishing criteria, you may recognize that you have not included a necessary domain expert. Or, your preliminary analysis of results may indicate missing evaluation criteria. With this process, you will amend your process and repeat the steps with the additional definitions. Any product evaluation is a complex process and must not be considered in isolation. The evaluation of the company that provides the product as well as how the product will integrate with other products in the organization is critical to a successful evaluation.

#### BENEFITS AND PITFALLS

Whatever evaluation approach you take in choosing a COTS-based product, you can learn from the experiences of others. There are services and tools that can guide you in a COTS evaluation. CM-SEI has developed CURE<sup>SM</sup> (COTS Usage Risk Evaluation) and also conducts product evaluation workshops to assist organizations in appropriately staffing and conducting COTS assessments (CM-SEI, 2004). This section will highlight some common failures as well as great benefits of COTS-based systems.

#### Vendor Hype?

Success is predicated upon a good relationship with your vendor. At a minimum, the vendor must view the component you purchase as core to their product line. You should also look at the vendor’s history in supporting prior releases simultaneously with the current product release. Determine your ability to influence your vendor’s product direction. While you might like to have every requirement met exactly, it is to your advantage to shape the COTS product with the

---

<sup>1</sup> Federal Geographic Data Committee

support of the vendor. You do not want to receive special versions of the product that are only provided to you. Significant benefits are gained when many customers are using the same software. More thorough testing increases product stability and proper functioning.

A COTS-based system does not eliminate the need to engineer your application integration (Carney, 1997). Similarly, COTS-based systems do not mean there is no development on your part. You need to be confident that your vendor has clearly identified the integration requirements and who is responsible for performing (and paying for) the integration.

Consider all aspects of the product when measuring product maturity. You may get what you want in the latest release, but has that release involved significant architectural changes that would reduce product stability – even when the vendor has a good history? Are you sold on buzzwords? Consider your reasoning when requiring adherence to any new technology. Is that technology really proven in a production environment? How will it benefit your users?

COTS-based systems must still be tested within the context of your usage. No one will have the exact same configuration that you do. And, changes in COTS-based products will have a ripple effect on your configuration. Plan and budget for some change with each release. Understand your vendor's policy for compatibility of APIs to prior releases. You can reduce the time you budget for testing because you have purchased COTS software, but you cannot eliminate test periods.

While “plug and play” has been fairly successful for hardware devices, it really hasn't happened yet for software. It takes work to make the pieces come together. More importantly, if you determine that the component you are using is not sufficient, it will take still more work to plug in an alternate component. Then, that component may exhibit significantly different semantic and presentation behaviors. If your COTS choice is based on the belief that you can swap out the component at a later time, you probably have not made the right choice.

Vendors will always emphasize the significant number of features that their software encompasses. But, in reality, more than half of the features in COTS products go unused. The wise consumer will look beyond the length of the list to the actual usability and applicability of those features.

### Real Benefits

Interestingly enough, the advantages of a COTS system can easily be viewed as disadvantages (CM-SEI, 2004). For example, on the one hand, you can identify predictable license costs as an advantage over the unknown costs for developing the software. Yet, you can also view licensing costs as a disadvantage – you must maintain the licenses to ensure upgrade releases.

By and large, COTS software will be better able to meet your scalability and stability requirements than custom developed software – in a shorter amount of time and for less cost. The COTS vendor typically has the advantage of greater depth of programming expertise than your IT organization will have and they are dedicated to the task of software development for commercial use. You also leverage the ideas of other users and their influence on the evolution of the COTS product. Deploying COTS software allows you to focus on YOUR business – not

on software development. The downside of this external influence is that the product is not tailored precisely to your business process.

### ROI

The motivation for COTS-based products is cost savings. But, while the development costs may realize such savings, what alternate costs are incurred? Workflows will change and skills required to manage the organization will change. COTS solutions are not just technological changes.

The new system may require data in a different format. If this data is passed from a legacy system, the cost to reformat that data may be prohibitive. Minimally, that cost must be factored into the evaluation.

You can benefit from standards developments and market trends that would not be obvious in a non-COTS-based system. You should expect that your vendor will track technology trends and fit those into the evolution of the product. While this is advantageous – you want to stay abreast of technology – it may require that you synchronize releases of more than one COTS component to leverage that technology. It can be argued that staying on current technology is an unnecessary risk for little advantage. However, at some point, you cannot sustain your business on unsupported versions of software and you will not be able to retain IT staff to maintain or enhance the older technologies.

In most cases, you will enjoy incremental system improvements for the same investment. You will get features you hadn't planned on. COTS decisions should be based on a complete life-cycle evaluation – requirements, design, implementation, test, deploy and maintain – not just one or two phases. Change will be constant and your organization must cope with that. Matching your system release to the marketplace realities is important to achieving any degree of success.

Post-deployment costs often exceed integration development costs (Basili, 2001). The key here is to budget for maintenance of your COTS-based product integration and to stay current on releases. Personnel factors are also critical when considering COTS – different skills are needed over custom software development. In a custom environment, your personnel are primarily traditional programmers. For a COTS solution, the staff often fills the role of integrator and innovator. They keep the project objectives in focus and work within the COTS environment and its extensions to devise a solution.

Project failure rates with COTS are higher than custom development but the benefits are higher as well. Several factors contribute to project failures:

- Mismatched skill on the evaluation and deployment teams
- Inexperience with COTS
- Overly optimistic expectations
- Overly aggressive deployment schedules
- Weak planning and managing of the project life cycle

Now, custom projects can fail for most of these same factors! As a project manager for any activity involving COTS, you can avoid failure by:

- Carefully choosing your team

- Setting appropriate expectations
  - Including users in your analysis and deployment plans
  - Building a strong relationship with your vendor
  - Recognizing that COTS is just one component of a well-planned and managed project.
- COTS projects should follow consistent and thorough life cycle planning just as custom development projects.

### One Size Fits All?

The change to a COTS-based system can be agonizing for an organization. The COTS products have inherent processes – and those may conflict with the user community. Part of your COTS evaluation must include process re-engineering. Changes in roles and responsibilities can reshape your organization.

Pilot programs and hands-on demonstrations with end-users are essential in the selection process. This activity will help you measure how close the COTS product matches your environment. You will also determine how much flexibility you have for adjusting the product. Some of the questions you should ask about configuration of your COTS product are:

- How much can you change the product?
- How much does that cost?
- How does this modification help achieve better integration and a better match to your workflow?

A COTS product that allows some degree of configuration is a much better choice than one that has no flexibility or one that requires significant additive programming.

## METADATA

### Metadata Types

At least two distinct types of metadata – data and application – exist related to GIS data. They both describe the geometry and attribution that represents physical objects or logical locations. The “data” metadata (Schweitzer, 2002) answers data questions:

- What does the dataset describe?
- Who, why and how was the dataset produced?
- How reliable is the data?
- How can someone get a copy of the data?

This “data” metadata is at the core of the Federal Geographic Data Committee (FGDC) initiative. This type of metadata allows organizations to provide data to support the National Geospatial Data Clearinghouse (NGDC). It also allows organizations to look for and incorporate this data into their own environments. While any GIS project needs data, this “data” metadata is not particularly important to COTS product selection and usage<sup>2</sup>.

Coulondre (Coulondre, 1998) proposes further extensions and usage of metadata to enable GIS applications. Metadata is divided into three levels of information.

- Description of the domain – terms and semantics applicable to the community of users for this data

---

<sup>2</sup> If the COTS product(s) require FGDC compliant metadata for the data, then you must factor that into any data acquisition.

- Characteristics of the exchanged data – data lineage, common practices for data production, data schema
- Geographical information – specific information about the data grouped by data set (information applies to the whole), category (class information) and object (instance information)

The author proposes that this classification, beyond the traditional FGDC and OGIS definitions, further enables users to effectively interpret and retrieve geographical data. This extension of metadata is only important in choosing a COTS product if a key component of your system is usage and exchange of data from multiple sources.

The second type of metadata – application – addresses the programmatic and/or behavioral aspects of viewing, creating and managing GIS data. Many sources explore knowledge or rule systems for controlling application behavior (Yialouris, 1997). Within a knowledge-based system, an administrator builds patterns or rules into the software application. The application then utilizes this information to guide the user within the application. Since the knowledge is entered by the organization that will use the application, the application appears customized for a specific set of users. To this end, a rule or knowledge-based COTS product can transcend the “one size fits all” limits commonly associated with COTS. Historically, a good rule-based system is expensive to build. This paper offers a simpler approach to application configurability for COTS products over the construction of a complex knowledge or rule system.

#### Application Flexibility

Extending the idea of data metadata, a COTS application can be built such that it interrogates a catalog or schema to understand the data it has and how to control that data. In fact, a metadata-driven application will utilize the “data” metadata. It will also have a wealth of parameters or configuration options to control the COTS product.

Specifically, through metadata, a COTS application will be much more flexible and it will allow the administrator to enhance the data model without changing the software. By matching the data model to your business model, you will have a better chance of succeeding with your COTS deployment. If the COTS application expects a configurable data model, the maintenance and operation of the system is not at risk as it would be if the COTS did not allow data model changes and custom changes were made by your vendor to accommodate this.

These aspects of a COTS product can be controlled by metadata.

- Data model
- Presentation and description of data
- Operation or command applicability to specific data
- Availability of commands to certain users
- Integrity rules for data management
- Command behavior
- Extension / integration points

If these aspects can be controlled without programming, the COTS product is more likely to meet the needs of your organization. Adjustable interface points within the COTS product enable the COTS product to better match your business needs and suit your users’ preferences.

## Examples

The following table briefly shows aspects of an application that might be controlled by metadata. The table describes a fixed system and a configurable (metadata-driven) system. In a fixed system, there is one and only one option for how the application behaves. In a metadata-driven system, each installation and configuration of the system allows site-specific behavior<sup>3</sup>.

<b>Application Feature</b>	<b>Fixed System</b>	<b>Metadata-driven System</b>
<b>Data Model</b>	Product defines the entity and attributes	Product provides a default definition but allows the configuration to identify an alternate definition or augment the default
<b>Presentation and description of data</b>	Product defines the number of attributes that describe the entity and the applicable symbol or style	Product allows the administrator to name the attribution and define the style
<b>Command applicability</b>	Product allows edits of all attributes and graphics	Product allows data to be identified as “read-only” or non-editable
<b>Availability of commands</b>	Product offers no distinction for user classifications or provides a pre-defined set of classifications that map to commands within the product	Product allows assignment of command functions to users at the granularity of the command
<b>Integrity rules</b>	Product defines a fixed set of integrity constraints	Product allows the integrity constraints to be defined per installation
<b>Command behavior</b>	Product command follows a fixed workflow	Product command relies on parameters to control the execution plan for the command
<b>Extension points</b>	Product provides a finite set of exchange formats and reports	Product provides access to data storage as well as free-format reporting

**Table 3 Controllable Application Behavior**

Metadata allows a COTS product to become unique to an organization. Yet, at the same time, the organization is still using a COTS product. This offers two distinct benefits for COTS that are often contradictory. The organization utilizes a common, well-tested software component. And, it achieves a better fit of the COTS product for the organization with less impact on the user workflows. The result – better user acceptance at less cost than custom programming.

## SUMMARY

As stated previously, COTS is ubiquitous and inevitable. Unfortunately, it is not the panacea once envisioned. But, it is also the reality of most businesses. The demands on GIS organizations to effectively and efficiently manage large volumes of data are high. In GIS, rarely is your priority the development of a tool for GIS data management – but your priority always includes leveraging that data to make effective business decisions. Do not base your COTS decision solely on the technology of data management. This aspect can be only one component in analyzing how the COTS product enables you to do your job more effectively. GIS-centric COTS products will offer the most value to your organization if you can readily fit them into the workflows and systems that support your business and staff. COTS products that include application metadata will inevitably adapt better to your business and users. Mapping COTS capability and configurability to your improving business processes and careful

---

<sup>3</sup> Even a configurable system has limits on what configuration it exposes. Any COTS analysis should carefully identify the extent and applicability of the configurable components.

cost/benefit analysis of integration, training, deployment and administration are key ingredients to a successful COTS project.

## REFERENCES

Basili, V., 2001, COTS-Based Systems Top 10 List: Computer, Volume 34, Issue 5, 91-93.

Carnegie Mellon Software Engineering Institute, 2004, COTS-Based Systems, on-line at <http://www.sei.cmu.edu/cbs/index.html>.

Carnegie Mellon Software Engineering Institute, 2004, COTS-Based Systems Lessons Learned Web Pages, on-line at <http://www.sei.cmu.edu/cbs/lessons/>.

Carney, D., 1997, Assembling Large Systems from COTS Components: Opportunities, Cautions, Complexities: SEI Monographs on the Use of Commercial Software in Government Systems, 1-14, on-line at <http://www.sei.cmu.edu/cbs/papers/monographs/assembling-systems/assembling.systems.pdf>.

Coulondre, Stephane, 1998, Metadata and GIS – A Classification of Metadata for GIS, 1-7, on-line at <http://citeseer.ist.psu.edu/cache/papers/cs/23466/http:zSzzSzwww.lirmm.frzSz~coulondrzSzpszSzGIS98.pdf/coulondre98metadata.pdf>.

Oberndorf, P., 2002, A Process for COTS Software Product Evaluation: Proceedings of the 1<sup>st</sup> International Conference on COTS-Based Software Systems, 86-96.

Priebe, Torsten, 2003, Towards Integrative Enterprise Knowledge Portals, 1-8, on-line at <http://citeseer.ist.psu.edu/cache/papers/cs/30910/http:zSzzSzwww.priebe-it.dezSzenzSz..zSzpublicationszSzPrPe03b.pdf/priebe03towards.pdf>.

Shengru Tu, L., 2002, Achieving Interoperability for Integration of Heterogeneous COTS Geographic Information Systems, Proceedings of the tenth ACM international symposium on Advances in geographic information systems, 162-167.

Sledge, C., 1998, Gotchas of COTS for Acquisition Managers: What You Don't Know Can Hurt You, on-line at [http://www.sei.cmu.edu/cbs/cbs\\_slides/stc98/Gotchas/index.htm](http://www.sei.cmu.edu/cbs/cbs_slides/stc98/Gotchas/index.htm).

Schweitzer, Peter N., 2002, GIS and Metadata, Easy as ABC – Putting Metadata in Plain Language, GEO World, online at <http://www.geoplace.com/gw/1998/0998/998abc.asp>.

Torchiano, Marco, 2002, COTS Products Characterization: Proceedings of the 14<sup>th</sup> international conference on Software engineering and knowledge engineering, 335- 338.

Voisard, Agnes, 1997, Abstraction and Decomposition in Open GIS, International Computer Science Institute, 1-22, online at <http://citeseer.ist.psu.edu/cache/papers/cs/3445/ftp:zSzzSzftp.icsi.berkeley.eduzSzpubzSztechreportszSz1997zSztr-97-006.pdf/voisard97abstraction.pdf>.

Yialouris, Constantine, 1997, An Integrated Expert Geographical Information System for Soil Suitability and Soil Evaluation, Journal of Geographic Information and Decision Analysis, Vol 1, No 2, 90-100.