

The Use of Profiling in XML Documents

Using profiling for personalized
information delivery

By Tonua G. Brown
Program Manager



Document Management Solutions, Inc.

48 Beechwood Drive, North Andover, MA 01845

Tel: 978.738.9770 Fax: 978.738.9765

www.dmsi-world.com

The Use of Profiling in XML Documents

What is profiling?

The dictionary definition of profile includes “a biographical essay presenting the subject’s most noteworthy characteristics and achievements.” This very closely describes the way we use profiles in business and, especially, e-commerce. Businesses assign profiles to customers that describe their needs, requirements, and interests. These profiles are then used to target the audience of particular products.

When you visit an e-commerce web site, like a music site that sells CDs, you provide them with your profile information when you register for the site. At the time you register, you tell them what kind of music you like, such as rock, country, jazz, or classical. Each time you make a purchase, your profile may be updated with the specific artists that you are purchasing. Some sites even have the ability to track your clicks through the web site to determine your interests. All of this information is combined to create your personal profile. Your profile is then used to target certain products at you. If you show an interest in certain types of music or even certain artists, you may start seeing advertisements when you visit the site that offer special deals on those artists’ recordings.

When we talk about profiling our data, we are referring to the way we mark up the data to indicate its target audience. Through the use of elements, XML allows us to mark up the data in such a way that the context of that data is described in the markup language itself. Additionally, attributes allow us to provide meta-data regarding that context. The target audience for the information is a type of meta-data and can be captured in attributes on each element. Therefore, by profiling our information according to its target audience, we can match the information profile to the audience profile in order to deliver the information that best meets the needs, requirements, and interests of our customers.

How are profiles assigned to data?

As described previously, there are many things that make up a customer’s profile. Consequently, there are many types of information profiles. Information can be targeted to a group, an individual, a characteristic of an individual, an output type, or even the product itself. All of these things can be combined in various ways to match specific customer profiles.

Profiling for groups

Information can be targeted at specific groups of users, even if we know nothing about the individual members of the group. For instance, a software product may

run on various operating systems: Unix, Windows, or Macintosh. When customers purchase the product, they choose the one for the operating system they use. They also expect to get installation and operating instructions that are appropriate for the chosen operating system.

At this point, we don't know anything about the individual customers, but we can already target the information they receive based on the operating system they use. This allows us to create a profile class. A profile class is a generic name for a category of profiles. In this instance, the profile class is operating system. The values of that class are Unix, Windows, and Macintosh. As we author the information, we can identify the target audience by assigning a value in a profile class.

Profiling for individuals

Once we identify the customer, we can now target information to that specific individual. It's important to understand that in this context the term individual most likely represents a group whose members have known requirements. When we refer to a customer as an individual, it is likely that the customer is a company that has many employees that will use the product and information. While it is possible to target information to an actual individual, such as "Mary McRae," it is unlikely that you will ever have a need to do so.

An example of profiling for a specific customer is if you develop customizations to your product that are available only to that customer. You want to document those customizations in addition to your standard product, but you don't want to make that information available to other customers. Therefore, you can assign a profile to the information about the customizations to target only the customer who received them. When the document is published for other customers the information about the customizations will be excluded.

An individual can also represent a role, such as engineer, programmer, or manager. Depending on the individual's role, he or she may get different information than someone in a different role. For example, if your product allows users to develop their own customizations, programmers will want detailed instructions for development. Managers may be interested in a high level overview of your development kit, but probably do not want to read the detailed instructions. Therefore, the detailed information can be assigned a profile for programmers. When the documentation is published for managers, they can get the information they need without having to wade through the detailed instructions that they are not interested in.

Profiling for characteristics

You can set profiles on characteristics of individuals, such as their skill level. In the telecom industry, it is standard practice to provide maintenance manuals according to the skill level of the field engineer. Experienced, or "expert," field engineers only require high-level information to instruct them on a maintenance procedure.

However, someone who is still considered a “novice” may require detailed information about each step in that procedure. Therefore, you can create a profile class for skill level and include expert and novice as its values. This enables the field engineers to get the right information based on their skill levels.

Profiling for output type

Profile classes can be based on the intended output of the information. For instance, you may want to include a form to allow users to request additional information. When you publish your document on paper, you have a mail-in form you want included. When you publish to the web, however, you simply want to include a hyperlink to an existing web form. Therefore, you can create a profile class for publish media. When authoring the information, the mail-in form would be assigned a value of paper. The hyperlink to the web form would be assigned a value of web. This enables you to deliver the right information to your customer based on the media by which they will access that information.

Profiling for products

When developing products, it is frequently important to differentiate the information associated with that product based on its release version. In addition, sometimes several software products may be based from the same core code. In these instances, while the documentation for each product may be similar, and in some areas even identical, it is important to distinguish what information is associated with what products. Therefore, you will want to profile your information based on the product with which it is associated. For instance, you may have an operating manual that you release with each version of your software. You may have a chapter that is appropriate for all versions of the software but contains paragraphs that are only appropriate for a specific version. Therefore, you can create a profile class for version with the software version numbers as the values for the class. This allows you maintain a single document that can be published for any version of the software.

How are profile classes related?

When assigning profiles to data, it's important to understand the relationships and dependencies among the profile classes. For example, you may create a profile class for hardware that includes the values PC, Sun, HP, and Macintosh. You may create a second profile class for operating system that includes the values of Windows 95/98, Windows NT, Unix, Linux, and Mac-OS. The values of the operating system class have dependencies on the values of the hardware class.

The dependencies among the profile classes affect how profiles are assigned to the data. It should be assumed that when a profile is assigned to an element, the entire content of the element automatically inherits that profile. Using the previous example, if you create a chapter that is profiled for Sun hardware, the entire

content of the chapter should be appropriate only to Sun hardware. Therefore, you would not want to—and should not be able to—include paragraphs in the chapter that are profiled for the Mac-OS operating system.

Inherit relationships vs. exclude relationships

When defining a single profile class, it is imperative that you also define the relationship that the values of the class have with one another. Is it an inherit relationship? Or is it an exclude relationship? In an inherit relationship, the values are hierarchical. Each value inherits the previous value. A security profile is a good example of an inherit relationship. A security profile class may include values like classified, secret, and top secret. Users who match the classified profile can see only classified information and nothing else. However, users who match the secret profile can see not only classified information, but also secret information. Likewise, users who match the top secret profile can see classified and secret information as well as top secret information. Therefore, when you profile your document for top secret, you do not want to exclude the classified and secret information. However, when you profile your document for classified, you do want to exclude the secret and top secret information.

In an exclude relationship, each value in the class stands on its own. The operating system profile class sample that we used previously is a good example of an exclude relationship. If you profile your document for Windows, you do not want to include any Unix or Macintosh information. Likewise, if you profile your document for Unix, you do not want to include any Windows or Macintosh information. Each value in the class excludes the other values. It is important to note that even in exclude relationships, you may have information that is appropriate for more than one value in the class while not being appropriate for all values in the class. In these cases, the information should be profiled for all appropriate values.

Publishing with profiles

When you are ready to publish your document, you are most likely going to publish against an information profile that contains several profile classes and possibly multiple values for each class. When determining what data is included in the output document, values within the same class will have an OR relationship. In other words, using the operating system class example, if your published document profile includes both Windows and Unix, an object will be included in the output document if it is profiled for either Windows OR Unix. Values for different profile classes will have an AND relationship. In other words, if your published document profile includes Sun and novice, then any information that is profiled for Sun and expert would be excluded from the published document. Objects that do not have any profile assigned to them are assumed to inherit all profiles. In other words, objects without an assigned profile are always included.

Information profile classes

In this example, we are creating a document that contains three profile classes that we can assign to our data. The following table shows the profile classes and each of their values.

| Profile Class | Values | | |
|-----------------------|---------|---------------|----------|
| Operating System (OS) | Windows | Unix | |
| User Role (Role) | Manager | Administrator | Engineer |
| Skill Level | Novice | Expert | |

Profile assignment

Throughout our document we have assigned profiles to objects. The objects and their assigned profiles are shown in the table below.

| Document Objects | | | Object Content | Assigned Profiles | |
|------------------|---------|-----------|--|--|----------------------|
| Chapter | Section | Paragraph | | | |
| 1 | | | Chapter applicable to Windows only. | OS = Windows | |
| | 1.1 | | | Section applicable to Windows and Engineer. | Role = Engineer |
| | | First | Paragraph applicable to Windows and Engineer and Novice. | | Skill Level = Novice |
| | | Second | Paragraph applicable to Windows and Engineer and Expert. | | Skill Level = Expert |
| | | Third | Paragraph applicable to Windows and Engineer and Novice and Expert. | | No profile assigned |
| | 1.2 | | | Section applicable to Windows and Administrator. | Role = Administrator |
| | 1.3 | | | Section applicable to Windows and Manager. | Role = Manager |
| 2 | | | Chapter applicable to Unix only. | OS = Unix | |
| | 2.1 | | | Section applicable to Unix and Engineer. | Role = Engineer |
| | | First | Paragraph applicable to Unix and Engineer and Novice. | | Skill Level = Novice |
| | | Second | Paragraph applicable to Unix and Engineer and Expert. | | Skill Level = Expert |
| | 2.2 | | | Section applicable to Unix and Administrator. | Role = Administrator |
| | 2.3 | | | Section applicable to Unix and Administrator. | Role = Manager |
| 2.4 | | | Section applicable to Unix and Engineer and Administrator and Manager and Novice and Expert. | No profile assigned | |
| 3 | | | Section applicable to Unix and Windows and Engineer and Administrator and Manager and Novice and Expert. | No profile assigned. | |

Published output

When an information profile is applied to a document for publishing, each object in the document is analyzed for inclusion or exclusion in the published document. Only

objects with profiles that match the applied information profile will be included. Remember that objects that have no profile assigned are assumed to match the profile. Also remember that objects will inherit the profile of its parent. Therefore, even if an object does not have an assigned profile, if its parent has an assigned profile, that object is considered to have the same profile as its parent. The following table shows the result of publishing the document in the previous table against the information profile that matches Windows, Unix, Engineer, and Expert. Note that this would be the same if the publishing profile matched Engineer and Expert and did not specify an operating system match.

| Document Objects | | | Object Content | Assigned Profiles |
|------------------|---------|-----------|--|----------------------|
| Chapter | Section | Paragraph | | |
| 1 | | | Chapter applicable to Windows only. | OS = Windows |
| | 1.1 | | Section applicable to Windows and Engineer. | Role = Engineer |
| | | Second | Paragraph applicable to Windows and Engineer and Expert. | Skill Level = Expert |
| | | Third | Paragraph applicable to Windows and Engineer and Novice and Expert. | No profile assigned |
| 2 | | | Chapter applicable to Unix only. | OS = Unix |
| | 2.1 | | Section applicable to Unix and Engineer. | Role = Engineer |
| | | Second | Paragraph applicable to Unix and Engineer and Expert. | Skill Level = Expert |
| | 2.4 | | Section applicable to Unix and Engineer and Administrator and Manager and Novice and Expert. | No profile assigned |
| 3 | | | Section applicable to Unix and Windows and Engineer and Administrator and Manager and Novice and Expert. | No profile assigned. |

Profiles applied in user environment

In the previous example, we showed a document that had been published for both Unix and Windows. If you are publishing your information to paper, it may be more appropriate to create separate documents for Windows and Unix. However, if you are publishing to a web site, for instance, it may be more appropriate to include both types of data in the same document. Then, when the user accesses the document on the web site, they can choose the information that is appropriate for them.

If you wish to make your information publicly available on a web site and do not know who will be accessing your site, it may be appropriate to allow users to select the profile that best meets their needs. All of the information that you wish to be publicly available would be contained in your document (or documents). When customers access your site, they could select options that best describe their needs. This would create profiles that would then be applied to the documents they access. However, if your information is placed on a secured site, you could associate profiles with customer logon IDs. This would allow customers to automatically be

fed the documents that match their customer profile. The profile would be applied at the time of access, but the customer would only see a document that is customized against his or her customer profile.

Publishing Methods

There are several methods for publishing profiled information. Your delivery method may dictate or influence your publishing method, as well. For instance, if you are publishing to paper, you must assemble your document prior to publishing. Once printed to the paper, it cannot be profiled further. However, if you are publishing to an extranet, you may want to allow dynamic document assembly based on the user profile associated with the logon ID of your customer.

Your publishing method may also depend on your methods for authoring and storing your information. Do you author and store whole documents? Do you author and store chapters, sections, or modules? Or do you store every element separately? If you are storing whole documents, your publish method will need to strip away the information that does not match the publish profile. If you are storing every element separately, your publish method will need to assemble the objects that match the publish profile and ignore those that do not. If you are storing your document objects in some chunk smaller than a whole document and larger than an individual element, then your publish method will need to include a combination of both stripping and assembly. You will first need to assemble all of the document objects that match the profile and then strip away the elements they contain that do not match the publish profile.

As soon as we begin discussion about assembling objects to create a document, the question is raised about whether it is more appropriate to store the profile information on the document objects or as meta-data in the document management system. This argument favors storing the information directly on the document objects rather than meta-data in the document management system. The reason for that is if you ever decide to change your document management system, you lose the meta-data. XML is application independent. Ideally, you should be able to switch applications in the middle of a project without affecting the data. Profiling information is part of the data. If you rely on your document management system to maintain the profile information, your data is no longer application independent. If you want to allow your document management system to do the assembly and it must reside in the meta-data in order to accomplish this, then it is recommended that you store the profile information on the document objects as well. Ideally, your document management system should be able to populate its own profile meta-data information based on the content of your profiling attributes.

Summary

In summary, profiling is a method for tracking the intended audience of your information. Profile classes define profiles categories. Each value in a class matches a specific profile for that category. Profiles can be combined to deliver highly customized information to your customers. Depending on your processes and delivery methods, you can choose to apply an information profile at the time you publish and deliver pre-assembled, tailored documents to your customers, or you can dynamically assemble the information when the customer accesses it by applying a profile that is associated with the customer's logon ID. Therefore, information profiles can be matched to customer profiles to provide your customers with information that best suits their needs.