AdsML® Framework for E-Commerce Business Standards for Advertising

Structured Descriptions of Advertisement Objects 1.0.8

Part 1

Usage Rules & Guidelines

Document Authors: AdsML Technical Working Group

Document ID: AdsMLStructuredDescriptions-1.0.8-SpecP1Usage-AS-9

Document File Name: AdsMLStructuredDescriptions-1.0-SpecP1Usage-AS.pdf

Document Status: Approved Specification

Document Date: 15 April 2010

Draft Number: 9

Table of Contents

1 AdsML Standard Documentation

1.1 Document status and copyright

This is the Approved Specification of *AdsML Structured Descriptions of Advertisement Objects 1.0 Part 1 Usage Rules & Guidelines*.

1.2 Non-Exclusive License Agreement for AdsML Consortium Specifications

USER LICENSE

IMPORTANT: AdsML Consortium specifications and related documents, whether the document be in a paper or electronic format, are made available to you subject to the terms stated below. Please read the following carefully.

1. All AdsML Consortium Copyrightable Works are licensed for use only on the condition that the users agree to this license, and this work has been provided according to such an agreement. Subject to these and other licensing requirements contained herein, you may, on a non-exclusive basis, use the Specification.

2. The AdsML Consortium openly provides this specification for voluntary use by individuals, partnerships, companies, corporations, organizations and any other entity for use at the entity's own risk. This disclaimer, license and release is intended to apply to the AdsML Consortium, its officers, directors, agents, representatives, members, contributors, affiliates, contractors, or coventurers (collectively the AdsML Consortium) acting jointly or severally.

3. This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this Usage License are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the AdsML Consortium, except as needed for the purpose of developing AdsML specifications, in which case the procedures for copyrights defined in the AdsML Process document must be followed, or as

required to translate it into languages other than English. The limited permissions granted above are perpetual and will not be revoked by AdsML or its successors or assigns.

4. Any use, duplication, distribution, or exploitation of the Specification in any manner is at your own risk.

5. NO WARRANTY, EXPRESSED OR IMPLIED, IS MADE REGARDING THE ACCURACY, ADEQUACY, COMPLETENESS, LEGALITY, RELIABILITY OR USEFULNESS OF ANY INFORMATION CONTAINED IN THIS DOCUMENT OR IN ANY SPECIFICATION OR OTHER PRODUCT OR SERVICE PRODUCED OR SPONSORED BY THE ADSML CONSORTIUM. THIS DOCUMENT AND THE INFORMATION CONTAINED HEREIN AND INCLUDED IN ANY SPECIFICATION OR OTHER PRODUCT OR SERVICE OF THE ADSML CONSORTIUM IS PROVIDED ON AN "AS IS" BASIS. THE ADSML CONSORTIUM DISCLAIMS ALL WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, ANY ACTUAL OR ASSERTED WARRANTY OF NON-INFRINGEMENT OF PROPRIETARY RIGHTS, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. NEITHER THE ADSML CONSORTIUM NOR ITS CONTRIBUTORS SHALL BE HELD LIABLE FOR ANY IMPROPER OR INCORRECT USE OF INFORMATION. NEITHER THE ADSML CONSORTIUM NOR ITS CONTRIBUTORS ASSUME ANY RESPONSIBILITY FOR ANYONE'S USE OF INFORMATION PROVIDED BY THE ADSML CONSORTIUM. IN NO EVENT SHALL THE ADSML CONSORTIUM OR ITS CONTRIBUTORS BE LIABLE TO ANYONE FOR DAMAGES OF ANY KIND, INCLUDING BUT NOT LIMITED TO, COMPENSATORY DAMAGES, LOST PROFITS, LOST DATA OR ANY FORM OF SPECIAL, INCIDENTAL, INDIRECT, CONSEQUENTIAL OR PUNITIVE DAMAGES OF ANY KIND WHETHER BASED ON BREACH OF CONTRACT OR WARRANTY, TORT, PRODUCT LIABILITY OR OTHERWISE.

6. The AdsML Consortium takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available. The AdsML Consortium does not represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication, assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification, can be obtained from the Secretariat of the AdsML Consortium.

7. By using this specification in any manner or for any purpose, you release the AdsML Consortium from all liabilities, claims, causes of action, allegations, losses, injuries, damages, or detriments of any nature arising from or relating to the use of the Specification or any portion thereof. You further agree not to file a lawsuit, make a claim, or take any other formal or informal legal action against the AdsML Consortium, resulting from your acquisition, use, duplication, distribution, or exploitation of the Specification or any portion thereof. Finally, you hereby agree that the AdsML Consortium is not liable for any direct, indirect, special or consequential damages arising from or relating to your acquisition, use, duplication, distribution, or exploitation of the Specification or any portion thereof.

8. This User License is perpetual subject to your conformance to the terms of this User License. The AdsML Consortium may terminate this User License

immediately upon your breach of this agreement and, upon such termination you will cease all use duplication, distribution, and/or exploitation in any manner of the Specification.

9. This User License reflects the entire agreement of the parties regarding the subject matter hereof and supercedes all prior agreements or representations regarding such matters, whether written or oral. To the extent any portion or provision of this User License is found to be illegal or unenforceable, then the remaining provisions of this User License will remain in full force and effect and the illegal or unenforceable provision will be construed to give it such effect as it may properly have that is consistent with the intentions of the parties. This User License may only be modified in writing signed by an authorized representative of the AdsML Consortium. This User License will be governed by the law of Darmstadt (Federal Republic of Germany), as such law is applied to contracts made and fully performed in Darmstadt (Federal Republic of Germany). Any disputes arising from or relating to this User License will be resolved in the courts of Darmstadt (Federal Republic of Germany). You consent to the jurisdiction of such courts over you and covenant not to assert before such courts any objection to proceeding in such forums.

10. Except as expressly provided herein, you may not use the name of the AdsML Consortium, or any of its marks, for any purpose without the prior consent of an authorized representative of the owner of such name or mark.

IF YOU DO NOT AGREE TO THESE TERMS PLEASE CEASE ALL USE OF THIS SPECIFICATION NOW. IF YOU HAVE ANY QUESTIONS ABOUT THESE TERMS, PLEASE CONTACT THE SECRETARIAT OF THE ADSML CONSORTIUM.

AS OF THE DATE OF THIS REVISION OF THE SPECIFICATION YOU MAY CONTACT THE AdsML Consortium at www.adsml.org.

1.3 AdsML Code of Conduct

The AdsML Code of Conduct governs AdsML Consortium activities. A reading or reference to the AdsML Code of Conduct begins every AdsML activity, whether a meeting of the AdsML Consortium, AdsML Working Groups, or AdsML conference calls to resolve a technical issue. The AdsML Code of Conduct says:

Trade associations are perfectly lawful organizations. However, since a trade association is, by definition, an organization of competitors, AdsML Consortium members must take precautions to ensure that we do not engage in activities which can be interpreted as violating anti-trust or other unfair competition laws.

For any activity which is deemed to unreasonably restrain trade, AdsML, its members and individual representatives may be subject to severe legal penalties, regardless of our otherwise beneficial objectives. It is important to realize, therefore, that an action that may seem to make "good business sense" can injure competition and therefore be prohibited under the antitrust or unfair competition laws.

To ensure that we conduct all meetings and gatherings in strict compliance with any such laws and agreements in any part of the world, the AdsML Code of Conduct is to be distributed and/or read aloud at all such gatherings.

- There shall be no discussion of rates, fares, surcharges, conditions, terms or prices of services, allocating or sharing of customers, or refusing to deal with a particular supplier or class of suppliers. Neither serious nor flippant remarks about such subjects will be permitted.
- AdsML shall not issue recommendations about any of the above subjects or distribute to its members any publication concerning such matters. No discussions that directly or indirectly fix purchase or selling prices may take place.
- There shall be no discussions of members' marketing, pricing or service plans.
- All AdsML related meetings shall be conducted in accordance with a previously prepared and distributed agenda.
- If you are uncomfortable about the direction that you believe a discussion is heading, you should say so promptly.

Members may have varying views about issues that AdsML deals with. They are encouraged to express themselves in AdsML activities. However, official AdsML communications to the public are the sole responsibility of the AdsML Consortium. To avoid creating confusion among the public, therefore, the Steering Committee must approve press releases and any other forms of official AdsML communications to the public before they are released.

1.4 Document Number and Location

This document, Document Number AdsMLStructuredDescriptions-1.0.8-SpecP1Usage-AS-9, is freely available. It is located at the AdsML website at http://www.adsml.org/.

1.5 Purpose of this document

This document provides a management-level overview of the AdsML Structured Descriptions of Advertisement Objects. It provides the reader interested in understanding the nature of the AdsML Structured Descriptions materials and how to use them with the necessary context and supporting information.

1.6 Audience

The intended audience for this document is any prospective user of the AdsML Structured Descriptions materials, interested parties, and the AdsML Consortium.

Comments on this document should be addressed to the Technical Working Group of the AdsML Consortium (technical.wg@adsml.org).

1.7 Accompanying documents

This document serves as the reference guide to the AdsML Structured Descriptions messages to address specific business requirements. A companion document, *AdsML Structured Descriptions of Advertisement Objects 1.0 Part 2 Specification & Schema*, provides additional rules and guidance for using AdsMLStructuredDescriptions schema. They are meant to be read together.

In addition, elements and structures that are used in multiple AdsML schemas are documented in the *AdsML Type Library* specification. AdsML Structured Descriptions makes extensive use of such structures, therefore the *Type Library* specification is an essential reference.

All three documents are part of the AdsML Framework, which contains a suite of related documents. Readers of this document are assumed to be familiar with the

full range of relevant AdsML documentation. In particular, readers are assumed to have read the *E-Commerce Usage Rules and Guidelines* document. A description of the entire document set can be found in the *ReadMeFirst* html file associated with this release of the Framework.

1.8 Change History

Draft	Date	Changes	Author
AS 1 1.0.0	14 September 2005	First Approved Specification. Document renamed from "Structured Descriptions Overview". Previous change history removed.	TS
AS 2 1.0.1	1 June 2006	New introduction section added.	JC
AS 3 1.0.2	1 October 2006	Maintenance version release; no significant changes. AdsML references changed to reflect registered trademark status.	JC
AS 4 1.0.3	1 October 2006	Aligned draft version number	TS
AS 5 1.0.4	1 October 2006	Updated to use Controlled Vocabularies 3.0. No other changes.	UW
AS 6 1.0.5	10 October 2007	Updated to use AdsML Type Library 2.0 and minor editorial revisions.	JC
AS 7 1.0.6	30 June 2009	Edited to incorporate introductory text that had previously been in the Framework Overview document	TS
AS 9 1.0.8	15 April 2010	Minor editorial revisions. Errata in minor versioning corrected.	JC

1.9 Acknowledgements

This document is a product of the AdsML Technical Working Group.

Primary authorship and editing was performed by:

- Tony Stewart (RivCom) – tony.stewart@rivcom.com
- Eli Rousso (AdStar) – eli@adstar.com
- Christian Rohrbach (iware – PubliGroupe)

Additional authorship and editing was performed by:

- Jay Cousins (RivCom) – jay.cousins@rivcom.com

Acknowledgements and thanks to other contributors for additional input to this document are listed in Appendix A: Acknowledgement for contributions to this document.

1.10 The AdsML Consortium

The documents comprising the AdsML standard were written by the AdsML Technical Working Group, a committee charged with creating the consortium's technical deliverables, and then approved by the entire membership.

More information about the consortium can be found on the consortium's website: www.adsml.org.

2 Introduction

The AdsML *Structured Descriptions of Advertisement Objects* standard defines an XML format and approach for describing in a structured, machine-processable way the objects that are featured in an advertisement. This includes both the products and services that are being offered or sought, and also secondary objects relevant to the advertisement such as the location of offering, keywords and the advertiser's reply-to details. The goal of this activity is to support the automated classification, syndication and aggregation of advertisements by multiple publishers, and then searching and querying of the contents of those ads by potential consumers. While the format is specifically required to be able to convey the description of items in classified ads, it is capable of describing the advertised items in any kind of advertisement in any medium.

This standard includes both an XML interchange format, and also a relatively non-technical mechanism by which domain experts can define controlled vocabularies of terms and measures (including constraints such as data types, value ranges, and lists of allowed values) that can be used to describe the features of specific types of things offered for sale in advertisements, for example, Cars or Houses. The mechanism includes tools and techniques to support both the business process of agreeing on the terms and constraints that will be used in a specific vocabulary, and the technical process of converting those terms and constraints into a structured AdsML vocabulary. Once an AdsML vocabulary has been created, it will be possible to use that vocabulary to support data-entry routines in ad order processing software (for example, by providing picklists of allowed values) and to validate that the information recorded about a given advertisement conforms to the governing vocabulary.

2.1 Model Vocabularies

The AdsML Technical Working Group has used these mechanisms to create several model AdsML vocabularies: Housing, Recruitment, Transportation, Travel, Miscellaneous Products and Miscellaneous Services. These AdsML vocabularies build on the information contained in the CREST 2.0 standard, while adding additional domain knowledge. We anticipate that in coming years, we will create model vocabularies to cover other common types of objects mentioned in classified ads.

Advertisers and publishers in a given country or region will be able either to modify the supplied AdsML model vocabularies so that they meet their needs, or to use the AdsML mechanisms defined in this project to create their required vocabularies from scratch.

2.2 Design approach

AdsML Structured Descriptions relies on earlier experience and standards that have been embraced and extended in order to support current advertising business requirements. In addition, AdsML Structured Descriptions has been designed with extensibility as an important objective in order to be able to grow with the business and support various business models and future requirements.

AdsML provides an XML framework, called the "AdsML Framework", for unifying and extending XML advertising standards. Where earlier advertising standards for e-commerce such as IfraAdConnexion or CREST focused on specific parts of the

overall advertising process, the AdsML specifications fill in the gaps between such standards and specifications, extend their reach and encourage convergence when they overlap. In this line of effort, the AdsMLStructuredDescriptions standard has been developed by the AdsML Consortium as the preferred approach for providing structured metadata describing advertisement content.

AdsMLStructuredDescriptions can be used standalone, but its main envisaged use is as a module 'plug in' used by other standards within the AdsML Framework. To this end, it is used by AdsMLMaterials and AdsMLBookings, for example. As the AdsML Framework evolves and further standards are developed by AdsML that require the structured description of advertisement objects then AdsMLStructuredDescriptions will be used by these standards.

3 Overview

3.1 Purpose

The primary purpose of the AdsML Structured Descriptions standard is to facilitate the process by which publishers and advertisers integrate their systems in order to transmit classified ads from one system to another. While this standard will be useful for other types of ads (display, broadcast, etc.), the initial business goal is to reduce the costs associated with the syndication and republishing of traditional, textually oriented classified ads in print and online media. The standard achieves this goal by providing direct support for the automated classification, syndication and aggregation of advertisements by multiple publishers, and then for the searching and querying of the contents of the published ads by potential consumers. Other business benefits, such as the ability to generate a visual representation of an ad based on its structured description, may be achieved in the process, but they are not the immediate goal.

See Appendix A: Excerpts from the formal requirements, for a further explanation of the vision of this project.

A "structured description of advertised objects" is basically an XML message that conveys in a well-defined way the information that was provided by the advertiser as part of the ad. Mostly this information will be about the advertised items themselves (e.g. Item=Car, Make=Ford, Model=Taurus, Body=Wagon, Model Year=2002, etc.), but it can also include reply-to details for the advertiser (e.g. advertiser's name, address, box number, email, phone, etc.), and non-printed information such as search keywords that should be associated with this ad in an interactive environment.

3.2 Process

The process that two (or more) trading partners go through when they decide to integrate their systems and exchange classified ads breaks down (at a very high level) into at least these steps:

1. Agree on the specific types of information that will be exchanged between them, including any business rules (such as mandatory vs. optional types of information)
2. Agree on the technical format that will be used to convey the information
3. Configure the upstream system to capture the necessary information (if it is not already present) and to validate mandatory business rules during data entry or capture
4. Configure all systems to export and import the agreed information in the agreed technical format

5. During testing, validate the messages to ensure that everything is working properly

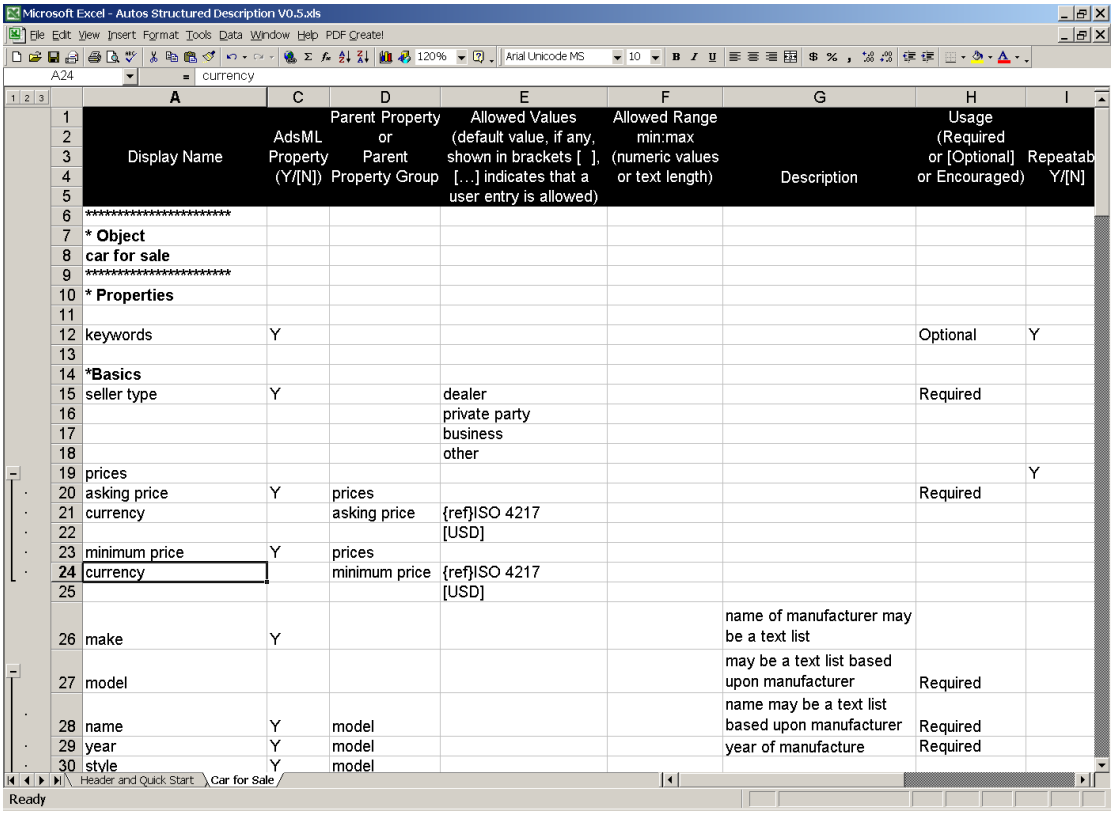
The AdsML Structured Descriptions standard is designed to facilitate all of these steps, as follows:

3.2.1 Agree on the specific types of information and business rules

We have provided a relatively simple form (spreadsheet) and rules for filling it in, which allows non-technical experts to agree on the nature of the information they need to exchange without getting bogged down in technical details. In effect, one can use this form to create a vocabulary about any type of advertised item – a car, a house, a job, even the all-important "other" category that doesn't fit into any of the usual buckets. In AdsML terminology, the vocabulary defined in this spreadsheet is called a "rule set".

Note that the spreadsheet provides one way to record a rule set, but there are others. For example, the Structured Descriptions standard also defines an XML format that can be used for a rule set.

By way of example, here is a screen shot of part of the form, showing the definition of a "Car for Sale":



Microsoft Excel - Autos Structured Description V0.5.xls

File Edit View Insert Format Tools Data Window Help PDF Create!

A24 = currency

	A	C	D	E	F	G	H	I
1	Display Name	AdsML Property (Y/[N])	Parent Property or Parent Property Group	Allowed Values (default value, if any, shown in brackets [ ], […] indicates that a user entry is allowed)	Allowed Range min:max (numeric values or text length)	Description	Usage (Required or [Optional] or Encouraged)	Repeatab Y/[N]
6	**********							
7	*** Object**							
8	**car for sale**							
9	**********							
10	*** Properties**							
11								
12	keywords	Y					Optional	Y
13								
14	***Basics**							
15	seller type	Y		dealer			Required	
16				private party				
17				business				
18				other				
19	prices							Y
20	asking price	Y	prices				Required	
21	currency		asking price	{ref}ISO 4217				
22				[USD]				
23	minimum price	Y	prices					
24	currency		minimum price	{ref}ISO 4217				
25				[USD]				
26	make	Y				name of manufacturer may be a text list		
27	model					may be a text list based upon manufacturer	Required	
28	name	Y	model			name may be a text list based upon manufacturer	Required	
29	year	Y	model			year of manufacture	Required	
30	style	Y	model					

Header and Quick Start | Car for Sale

Ready

In this screen shot we see that someone has defined the properties of a Car for Sale to include:

- Keywords (the Repeatable column on the far right indicates that the structured description can contain as many Keywords as necessary)
- Seller Type (the Usage column indicates that this field is required, and the Allowed Values column indicates that its value must come from the provided list of: dealer, private party, business, or other)
- Prices – a repeatable group containing two properties:

    - Asking Price (accompanied by a Currency modifier whose value must be taken from the ISO 4217 list of currency codes, with a default of "USD")
    - Minimum Price (also accompanied by a Currency)
- Make (text)
- Model – a group of properties including:
    - Name (text)
    - Year (text)
    - Style (text)
    - … and others that we cannot see in this screen shot

Other columns that don't appear in the screen shot include the ability to specify a data entry mask (e.g. "99.99") and the number of decimals, and whether or not a property is Publishable. The set of rules that can be specified is relatively limited, yet we believe they conform to the 80/20 principle by allowing advertisers and publishers to define all of the information they typically need to exchange, with minimal complexity.

Besides the blank template and instructions, we have provided several AdsML "model vocabularies" - versions of the form (as above) that have been filled in with sample data so that users can see how to use the form in practice. Users can also modify the sample data in order to create their own vocabularies.

We expect that use of this form and the model vocabularies will speed the process by which companies agree on the data that will be exchanged between their systems and the rule sets that will govern that data. However, use of the form is optional, and most of the value of the Structured Descriptions initiative comes from the following steps.

3.2.2 Agree on the technical format that will be used

Effectively, this step is going to go away.

AdsML will standardize this process so that once a rule set for a type of information has been defined (whether by filling in the spreadsheet or by any other means) it will figuratively be a matter of "pushing a button" to generate both an XML document that contains the definitions and rules that were entered in the form, and a standardized XML format for conveying this type of information between systems.

3.2.3 Configure the upstream system to capture the necessary information and validate mandatory business rules

AdsML cannot standardize the implementation of this step. It depends on the vendor of the system in which the ad information is captured. However, by providing a standard format (the spreadsheet and its XML equivalent) for representing both the types of information that will be captured and the types of business rules that are commonly enforced, we expect to:

A. Improve the flow of information to the developer who is configuring the screens or tools in which the advertising information is captured.

B. Enable vendors to develop systems that are optimized to support this configuration process. For example, pick lists of possible values for a field could be pulled directly from the XML version of the rule set, eliminating

the need to re-key those lists into a local database or (worse) into source code.

C. Conceivably, enable vendors to develop systems that can "read" the XML version of the rule set and use that information to auto-generate the entire data entry form with little or no manual customization.

3.2.4 Configure their systems to export and import the agreed information in the agreed technical format

This step should be greatly reduced. The AdsML Structured Descriptions standard will provide vendors with a known XML format that they need to export and import, and a known process by which to map the information being exchanged to the contents of their local databases. In most cases, they will still need to do this mapping, but at least the AdsML standard will have pinned down as many variables in this process as possible, so that vendors can provide tools that are optimized around those known points.

3.2.5 During testing, validate the messages to ensure that everything is working properly

This step will be handled automatically. The standard includes a set of procedures for converting an incoming XML message into a format that can be validated by XML tools, and for generating an XML Schema that will perform those validations according to the rules that have been agreed by the trading partners. The normal workflow is optimized for straight-through processing of trusted messages, and the validation of those incoming messages is treated as an optional step during which throughput is not so critical.

3.3 Assumptions and approach

The following assumptions govern the technical and process choices that have been made in this project.

1. It will not be possible to "standardize" the types of information (including property names) that will be exchanged between systems.

   While earlier efforts in this area (e.g. CREST) incorporated full-fledged vocabularies in their standards and attempted to get everyone to agree to use these vocabularies in their messages, this has not worked out in practice. And there is no possibility of achieving such a unified vocabulary across the globe. Therefore, the AdsML approach is to acknowledge that any group of trading partners will want to customize or create their own vocabulary of terms, and to provide tools and techniques that facilitate this process.

2. System vendors and implementers would prefer to deal with a single standardized XML format that can accommodate all of these customized vocabularies, so that they can build generic modules for message processing and data storage.

   The approach we have taken is to define an XML format that uses generic element names like <Property> and <Category>, in which the vocabulary information is provided with attributes, e.g. <Category type="car"> and <Property type="Make">Ford</Property>. (Note that these are not real examples of the syntax.)

3. Message validation is most important when integrating with a new trading partner. Once both systems are tested and in production, speed and simplicity of message processing is more important than validating each incoming message.

   We have optimized the message format for simplicity of processing, but also providing a standardized mechanism by which a system can take a few extra steps to validate that an incoming message conforms to the agreed rule set.

3.4 Illustrative usage scenario

1) Two or more companies in a given region decide to implement digital exchange of classified advertisement information between their systems. For example, perhaps one company will provide a software interface by which ad buyers will self-book classified ads, while a group of companies will publish the ads that have been booked. The requirement in this example is to transmit the ad booking information, including structured descriptions of the objects offered for sale in those ads, from the system in which it was booked, to the relevant publishers.

2) Representatives of the companies get together in order to agree on the information that needs to be exchanged between their systems. For example, if only car ads will be booked in this system, they agree on the properties of a car that users will be allowed or required to enter during the booking process (e.g. its price, make, model, age, identification number, etc.).

3) To facilitate this discussion, they use a copy of the spreadsheet containing the AdsML "Transportation" model vocabulary, which was provided as part of the AdsML standard, that has been filled in (by the Consortium) with commonly used properties of Cars, Trucks, Motorcycles, Boats, etc. They edit the information in this spreadsheet (deleting, renaming and/or adding classes and properties as necessary) until it reflects the information that will be exchanged between their systems. During this process they assign a unique name to this new rule set that they have created.

4) A developer working for one of the companies follows the rules defined in the AdsML standard to convert the information in the spreadsheet into a "machine readable" rule set and message format for conveying the agreed set of information.

5) The company providing the classified ad booking software configures its system to generate this XML format.

6) The companies receiving the classified ad bookings configure their systems to receive this XML format.

7) During testing, each company uses the machine-readable rule set to provide front-line validation of such basic features of the messages as their structure and whether all properties that were flagged as "required" on the spreadsheet are actually present in the incoming data.

   a) Note: in order to perform these validations, the companies use an XSLT stylesheet to transform the machine-readable rule set into a formal XML Schema, and then use another XSLT stylesheet to transform the structured descriptions contained in each set of advertising content into an XML message format that is designed to be validated by that XML Schema.

They are then able to use standard XML Schema validation tools to ensure that their systems are correctly generating the structured descriptions. Although this validation process is a little slow, it has the advantage of being performed entirely with off-the-shelf XML tools, and it is only done during the system setup phase.

8) Once the companies are satisfied that their systems are performing correctly, they turn off this type of validation.

3.5 One Rule Set, one human language

The AdsML Framework allows human readable textual metadata in AdsML messages in more than one human language: elements containing human readable text such as descriptions can be repeated, supporting multilingual workflows.

AdsML Structured Descriptions is by design monolingual. That is, a rule set defining an ad object's properties and structured descriptions data describing an instance of that ad object will always be in one human language.

3.5.1 Usage Rules & Guidelines

- If different human language versions of an entire Rule Set are required, then a Rule Set **SHOULD** be created for each different language version. The human language of the rule set **SHOULD** be recorded in the rule set's header and the display names for properties **SHOULD** be provided in that language. In the XML representation the language **SHOULD** be recorded using the *xml:lang* attribute of the `ObjectDefinition` element. Each different language Rule Set **SHOULD** be defined as a different `ObjectDefinition` element.
- If different human language versions of the values used in a Rule Set are required, then a Value Table **SHOULD** be created for each different language version. In the XML representation the language **SHOULD** be recorded using the *xml:lang* attribute of the `ValueTable` element.
- The XML schema for Structured Descriptions uses the `adsml:Description` element. This element has optional *i18nAttributes* that can be used to identify the human language of the description's text. In the AdsML Structured Descriptions context these elements **SHOULD NOT** be used: as described, AdsML Structured Descriptions is monolingual; the human language used is only specified once at the header level of a Rule Set or Value Table.
- Although human language can be specified in a Rule Set or Value Table definition, it cannot be recorded in instance AdsML `StructuredDescriptions` data. This is because `StructuredDescriptions` are used within `AdContent` and ad content is always monolingual. If there are different human language versions of an ad, then these **MUST** be treated as different sets of ad content with their own instance Structured Descriptions data. If the human language of the ad content needs to be recorded, then it **SHOULD** be in the ad content metadata – for example, as a property.

4 Rule Sets

This section provides a more detailed explanation of how to create a Structured Descriptions "rule set" using the supplied spreadsheet interface.

4.1 Key concepts

4.1.1 Rule sets

The AdsML "structured description" standard provides both a methodology and a technical format for conveying, in a well-defined way, the information that is provided by the advertiser as part of the ad. Mostly this information will be about the contents of the advertisement itself, but it can also include reply-to details for the advertiser, non-printed information such as search keywords that should be associated with this ad in an online environment, and other information that supports the placement and/or viewing of the ad. This type of information is usually referred to as "ad metadata".

A "Rule Set" is a document that supports the listing of all the ad metadata items used to describe a specific type of ad. The AdsML Structured Descriptions standard defines both spreadsheet and XML versions of rule sets. The spreadsheet version is easier to read and maintain; the XML version is able to support slightly more complex technical capabilities.

In this section of the Overview, we describe how to use the spreadsheet interface to define an AdsML Structured Description rule set. Information about the additional capabilities of the XML format can be found in the Structured Descriptions Specification.

4.1.1.1 Rule set conformance

Note that users do not need to use the spreadsheet interface in order to conform to the AdsML Structured Descriptions standard. The spreadsheet mechanism is provided because it is a more convenient format than an XML document in which to define the structure of the ad metadata.

In order to conform to the Structured Descriptions standard, trading partners must:

a) Define the information they wish to exchange in one or more XML rule sets that conform to the AdsML Structured Descriptions Standard

b) Exchange the information in XML instance messages, where the structure of each message conforms to the AdsML Structured Descriptions Standard, and the contents of each message conforms to one or more of the XML rule sets that were created in the previous step.

Further conformance requirements can be found in the Specification.

4.1.2 Objects and properties

Each type of object or service that can appear in an ad is called an "Advertisement Object". Since an advertisement will often consist of several different types of things (e.g. the products or services for sale, the advertiser's name and reply-to information, possibly other information), a Rule Set contains the rules for defining all of the different things that may appear in the type of ad to which it applies. Each object definition is recorded on its own page of the spreadsheet.

(Note: We are aware that the name "Advertisement Object" may sound overly technical for many users, but we have not been able to come up with a better one that is equally accurate.)

The metadata items themselves are referred to as "Properties" (of the advertisement object to which they apply). The objects and their properties can be thought of as a set of rules for the metadata that describes a particular type of advertisement; hence the name "Rule Set".

A defined rule set is unique (it is known by a Unique Identifier) and contains one or more advertisement objects, each with its specific set of metadata. The unique rule set may be one that is published and publicly available (such as the samples provided by the AdsML Consortium), one that is a trading partner-modified version of a publicly available rule set (and since it differs from the original, has its own Unique Id), or one that is privately agreed among trading partners (and also has a Unique Id).

4.1.3 Attributes and modifiers

Each property within a rule set has a set of attributes to specify things like its data type, data format, allowed values and business rules. These are recorded in columns of the spreadsheet. The columns are arranged so that the attributes that are most "business significant" are placed to the left, and the more technical columns are placed to the right.

Within a rule set, some properties may be grouped together as a logical unit referred to as a "Property Group". For example, a car engine may be described by a group of properties such as fuel type, displacement, horsepower, torque, etc. Similarly, a room may be described by a group of properties such as Name (e.g. "bedroom"), Dimensions, Number-of-closets, etc. The use of property groups is optional and is used for clarity. They are especially useful to represent a set of grouped properties that are repeatable – for example, the rooms in a house. You can define the property group just once in the rule set and mark it as repeatable, then repeat it as often as necessary in an instance message to describe, for example, the Bedroom, Living Room, Kitchen, etc.

Properties may have one or more "Modifiers". A modifier is used to further specify the attributes of a property. For example, the asking price for a car is specified as a decimal number, and its unit of currency is a modifier of the asking price that tells us how to value that number.

4.2 Spreadsheet format

The following notes provide basic information and rules for developing and/or using the Rule Set spreadsheet. Much of this information also appears in the first page ("Header and Quick Start") of the spreadsheet.

4.2.1 Page structure

Each Rule Set spreadsheet is divided into multiple pages: a single "Header and Quick Start" page, followed by one or more Object pages, each of which contains the rules for a specific type of Advertisement Object.

In general, users are expected to create rule sets that group together the definitions of similar objects. For example, a Transportation rule set might contain definitions for Cars, Trucks, Motorcycles, Boats, etc. Each of these would be on its own page of the Transportation spreadsheet.

The Rule Set Header is the identifier of the particular Rule Set. It contains information about the Rule Set and not ad metadata. It appears on the first page of the spreadsheet, along with some explanatory "quick start" information.

Each spreadsheet page other than the Header page is used to define a single Advertisement Object.

4.2.2 Rows and Columns

On each page other than the header, each spreadsheet row is used to list a Property, a Property Group or a Modifier, for example:

	A	B	C	D	E	F	G	H
1	Display Name	Type (Object, [Property], Group or Modifier)	AdsML Property (Y/[N])	Parent Property or Parent Property Group	Allowed Values (default value, if any, shown in brackets [ ], […] indicates that a user entry is allowed)	Allowed Range min:max (numeric values or text length)	Description	Usage (Required or [Optional or Encourage
40	exterior color	property	Y					Required
41	interior color	property	Y					
42	doors	property	Y				number of doors	
43	new	property	Y				new car	
44	warranty	property	Y				description	
45	location	property	Y				where located	
46	mileage	property	Y			0:999999	current mileage	Required
47	uom	modifier		mileage	[miles]			
48					kilometers			
49	condition	property	Y		[…]			Encouraged
50					as new			
51					excellent			
52					very good			
53					good			
54					fair			
55					poor			
56					damaged			
57					not working			
58					for parts only			
59	identification number	property	Y				vin (US) or equivalent	

The columns represent descriptive information about the Properties, and are labelled as follows:

A. **Display Name** – A descriptive name for the item described by this row of the spreadsheet. It is meant to be a human readable name containing spaces and whatever punctuation is necessary for clarity. Display Names preceded with an asterisk (*) are for convenience only when reading the spreadsheet - not part of the data.

Example: the word "Basics" is a label, not part of the rule set.

	Display Name	Type (Object, [Property], Group or Modifier)	AdsML Property (Y/[N])	Parent Proper or Parent Property Grou
14				
15	***Basics**			
16	seller type	property	Y	
17				
18				
19				
20	prices	group		
21	asking price	property	Y	prices
22	currency	modifier		asking price

B. **Type** – specifies the nature of the item described by this row of the spreadsheet: Object, Property, Property Group or Modifier. Defaults to "Property" if left blank.

C. **AdsML Property** – Indicates whether this property has been defined by the AdsML Consortium as part of a supplied model vocabulary. This is meant to facilitate interoperability, by differentiating the properties that are essentially unchanged – the way AdsML supplied them – from those properties that have been added or changed by trading partners in order to meet their specific requirements. This value should be set to "N" (or left blank" for any properties that you add to the vocabulary. Note, however, that if a property is flagged as "Required" in an AdsML model vocabulary, you may not delete that property or change its "AdsML Property" flag to "N".

D. **Parent Property or Parent Property Group** – specifies the name of the Property for which this is a Modifier (the Parent Property) or the name

of the Property Group that this Property or Property Group belongs to (if any). Mandatory for modifiers, and for properties or property groups that are part of another property group. A blank entry indicates that the current item is a Property or Object that does not have a parent.

Example: a "Prices" group contains two properties, "asking price" and "minimum price", each of which has a "currency" modifier. Each property lists "prices" as its parent property group, while each modifier lists its parent property.

A	B	C
Display Name	Type (Object, [Property], Group or Modifier)	Parent Property or Parent Property Group
prices	group	
asking price	property	prices
currency	modifier	asking price
minimum price	property	prices
currency	modifier	minimum price

E. **Allowed Values** – A list of allowed values for this property or attribute that can be used for one of two purposes: as a "locked down" list from which the value MUST come; or as a set of suggested values from which the advertiser will be encouraged to make a selection. See "Enumerated lists of allowed values", below.

F. **Allowed Range** – the minimum and maximum lengths for a text item or the minimum and maximum values for a numeric ("decimal" or "integer" data types) value.

G. **Description** – a descriptive phrase used to clarify the definition of the item when the Display Name and its attributes are not sufficient.

H. **Usage** – specifies whether supplied data about the Property is required, optional or optional but encouraged.

I. **Repeatable** – used to identify a Property or Property Group that is described once in this rule set but can be used multiple times in the metadata for an advertisement. For example, a Real Estate Rule Set could define a Repeatable Group of properties called "Room", which could then be repeated as often as necessary in a given message in order to define all the rooms in the house being advertised.

J. **Searchable** – specifies if the Property is to be searchable in interactive environments.

K. **Publishable** - specifies whether the Property may be published as part of the advertisement or must be kept private.

L. **Data Type** – allowed types are:

i. Text

ii. Date

iii. Date/Time

iv. Integer

v. Decimal – the number of decimals is specified in the Format Mask column.

vi. Y/N – shorthand for a "Boolean" entry whose value must be either Yes or No.

M. **Format Mask** - a data entry format "mask" or "regular expression" (e.g. "99,999.99" or "[A-C][F-Z]") used to constrain the values. Mandatory for data type Decimal in order to specify the number of decimals.

N. **Machine Name** – the name used internally for the XML when this data is transmitted, following XML Name rules. See additional naming rules below.

4.3 Rules and guidelines

4.3.1 Default values

Within the spreadsheet, where there is choice (as in the entry for an attribute, or a data value) a set of square brackets […] specifies the default value (if any).

4.3.2 Enumerated lists of allowed values

Enumerated lists of allowed values are recorded one item to a row in the Allowed Values spreadsheet column. A reference to an external list is signified by the {ref} precedent in the Allowed Values column; example {ref}ISO 4217 refers to the list of currency designators specified in ISO 4217.

If one of the list items is the default, then it is shown in the next row surrounded by square brackets; example [USD] specifies US Dollars as the default currency.

These capabilities can be combined, for example, by specifying both an external list and also a default value.

Example – the Currency modifier references an external list of ISO currency codes and also has a default value of "USD"

A	B	C	D
Display Name	Type (Object, [Property], Group or Modifier)	Parent Property or Parent Property Group	Allowed Values (default value, if any, shown in brackets [ ], […] indicates that a user entry is allowed)
prices	group		
asking price	property	price	
currency	modifier	asking price	{ref}ISO 4217
			[USD]
minimum price	property	price	

If the user is allowed to ignore the list of allowed values and enter whatever they want, then the string [...] is placed as either the first or last entry in the list.

Example – a list of suggested values for the "body style" property, with no default specified. The […] entry in the first row indicates that the advertiser may select from this list or enter something else.

A	B	C	D
Display Name	Type (Object, [Property], Group or Modifier)	Parent Property or Parent Property Group	Allowed Values (default value, if any, shown in brackets [ ], […] indicates that a user entry is allowed)
body style	property		[…]
			sedan
			coupe
			convertible
			wagon
			suv
			cab
			van

4.3.3 Grouping

All of the properties for a type of advertisement object (such as "car for sale") must appear together on a single page of the spreadsheet (except if they are pulled in by reference to another page via an external reference).

4.3.4 Keywords

By convention, the first item listed in an advertisement object's definition (i.e. on its page of the spreadsheet) should be an optional, repeatable and searchable property labeled “Keyword”. This is a free form entry that can be used to assist placing the advertised item in another, as yet unnamed, publication or media, downstream from the original publication of the ad.

4.3.5 Naming

4.3.5.1 Singular vs. plural names

In general, property and property group names should be singular, except that if a property or group is flagged as repeatable, its display name should be plural. However, this is just a guideline because there are many exceptions.

4.3.5.2 Unique names

Each Advertisement Object must be uniquely named within its Rule Set.

The Machine Name of a property or property group must be unique within the set of properties or property groups that share the same parent object or parent property group.

> For example, assume that property group “prices” contains two properties: “asking price” and “minimum price”. Because these properties are in the same property group, their machine names must be different from each other. Similarly, all of the properties that are in the “root” of a page of a rule set (i.e. they do not belong to a property group) are considered to have the same parent and therefore must be given unique machine names.

The Machine Name of a modifier must be unique within the set of modifiers that have the same parent property.

4.3.6 Property group attributes

Property Group attributes (columns) are restricted to Display Name, Machine Name, Parent Property or Group, and Repeatiblity.

However, if the definition of this property group is to be pulled in via an external reference to another object definition in the rule set (another page of the spreadsheet), then the name of that other object (spreadsheet page) is placed in the Data Type column.

4.3.7 Property group nesting

A property group can contain a mixture of properties and/or nested property groups. Nesting is designed to accommodate situations such as a house having Rooms (a repeatable property group), each of which can contain one or more Closets (a repeatable property group nested inside of the Rooms group).

A nested property group (that is, one contained inside of another property group) must have a different name from any of its parents or grandparents (etc.). Therefore, you cannot recursively nest a group inside of itself.

4.3.8 Re-use of object definitions

The entry for a property Group is normally followed by the list of properties and modifiers that are in that group, but it can optionally consist instead of a reference to another Advertisement Object that is defined on a different page of this rule set. Such a reference indicates that the definition of the other advertisement object should be copied in as the definition of this property Group.

This feature allows you to define common sets of properties that appear in many different objects. Put each common set of properties on its own page of the spreadsheet, and then reference them from all of the objects that need to include them. The reference is accomplished by typing the name of the page of this spreadsheet containing the object definition that you wish to reference in the Data Type column of the property Group definition.

Example – the "telephone", "fax" and "pager" groups are all defined using a data type of "telcom_number", which is not a real data type but rather the name of another page of this spreadsheet where their property definitions can be found.

	Display Name	Type (Object, [Property], Group or Modifier)	Data Type
48	telephone	group	telcom_number
49	mobile	group	mobile_telcom_numbe
50	fax	group	telcom_number
51	pager	group	telcom_number
52	instant messaging	group	Email address

4.3.9 Sequences

A property must be followed immediately by its modifiers, if it has any.

Within a property group, you must first list all of the Properties that are members of the group, followed by any other property groups that are nested inside it.

4.4 Modifying an AdsML model vocabulary

In general, the AdsML model vocabularies are intended to be used as a starting point. In most cases, trading partners will modify them to suit their specific business requirements.

In order to promote interoperability, however, it is important to maintain a level of control over how these vocabularies can be modified, and once a vocabulary has been modified, how to differentiate the original contents of the vocabulary from the added or changed properties.

If you plan to make substantial changes, you should rename the entire rule set so that it no longer references "AdsML" in its name, and make sure that none of the machine names of the items in the XML version of your rule set begin with the keyword "AdsML". Having done this, you are free to make whatever modifications you like, as if you were creating the vocabulary from scratch

Similarly, if you plan to make substantial changes to a property, property group or modifier in one of the model vocabularies, you should set its "AdsML Property" flag to "N" so as to indicate that it no longer conforms to the original AdsML-provided definition of that property. However, you are not allowed to do this if the property, property group or modifier in question is marked as "Required".

Otherwise, the following rules must be followed when editing an AdsML-supplied model vocabulary. These are copied from the original Requirements document for this project, so they are in somewhat formal language.

4.4.1 Allowed Values

Users **SHALL NOT** be allowed to change the Allowed Values or Range of any AdsML-provided property or modifier.

4.4.2 Extension

Users **SHALL** be allowed to add their own properties or modifiers to the vocabulary, and thereafter to modify any aspect of the properties and modifiers that they have added.

4.4.3 Regular Expression or Format

Users **SHALL** be allowed to change the Regular Expression or Format for any AdsML-provided property or modifier.

4.4.4 Renaming

Users **SHALL** be allowed to rename the Display name (but not the Machine name) of any AdsML-provided property or modifier.

4.4.5 Restriction

Users **SHALL** be allowed to remove any AdsML-provided property or modifier from an AdsML-supplied vocabulary except for those whose Usage is "Required".

4.4.6 Structural characteristics

Users **SHALL NOT** be allowed to change the Machine name, Property Type, Parent Property or Data Type attributes of any AdsML-provided property or modifier.

4.4.7 Usage characteristics

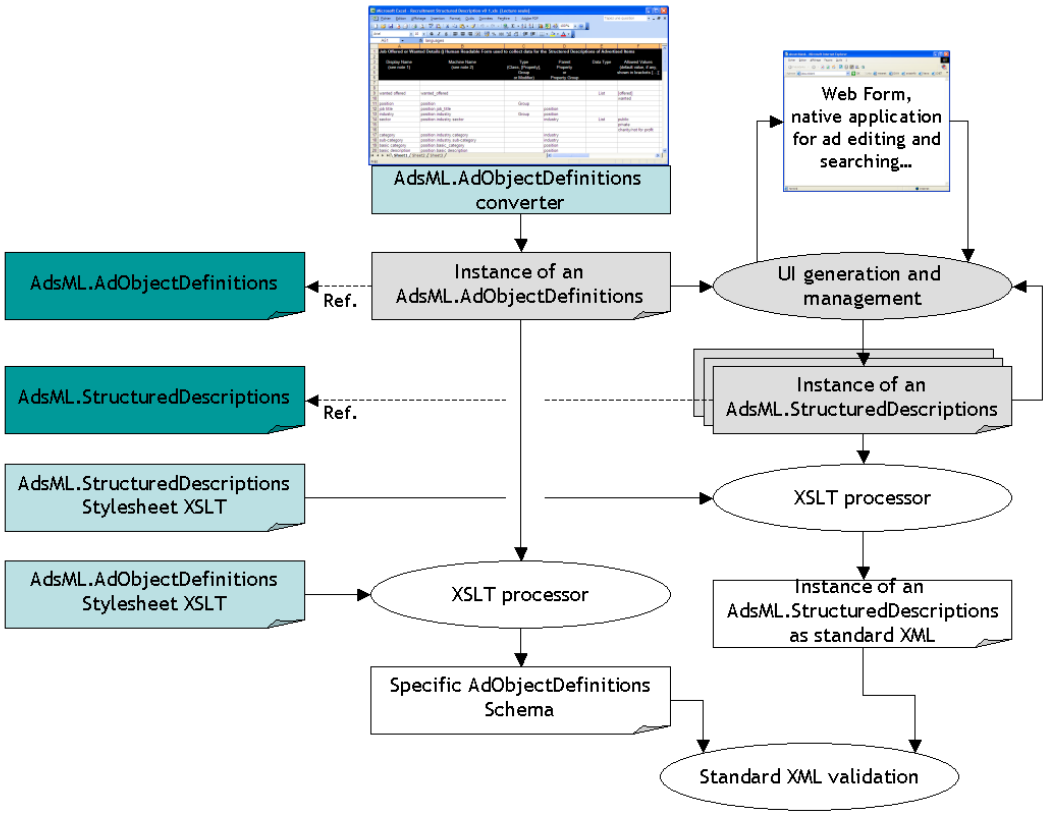
Users **SHALL** be allowed to change the Usage, Publishable, Searchable and Description attributes of any AdsML-provided property or modifier, except that

users **SHALL NOT** be allowed to change the Usage of a property or modifier whose Usage is "Required".

5 Architecture

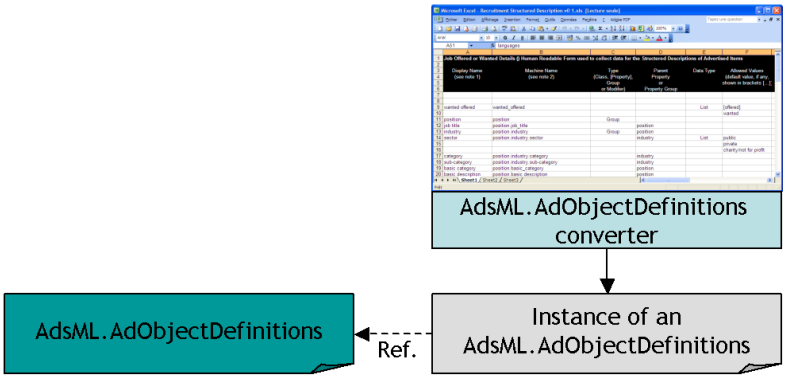
This section looks at the usage scenario from a more technical perspective, and discusses the process by which the information entered in a rule set is transformed into XML documents that are used to assist in the process of capturing, transmitting and validating structured descriptions of advertisement objects.

Below is an illustration of a model technical architecture showing the relationships between all of the main components, including both those defined in the AdsML Structured Definitions standard, and those that will be provided by implementers. The following sections then walk through five technical scenarios, highlighting and discussing the portions of this illustration that are relevant to each scenario.



5.1 Architectural views

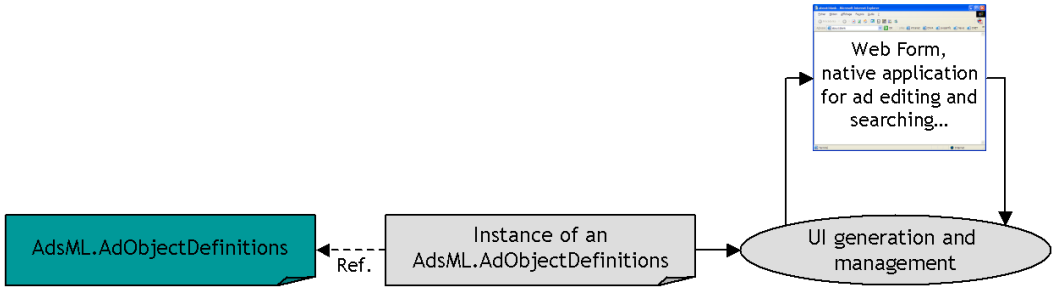
5.1.1 User's view



A content specialist defines the data structures and business rules that will be exchanged between these systems using the Rule Set spreadsheet interface that comes from AdsML. The spreadsheet is then converted into an AdsML.AdObjectDefinitions instance, which conforms to the structure defined in the AdsML.AdObjectDefinitions schema.

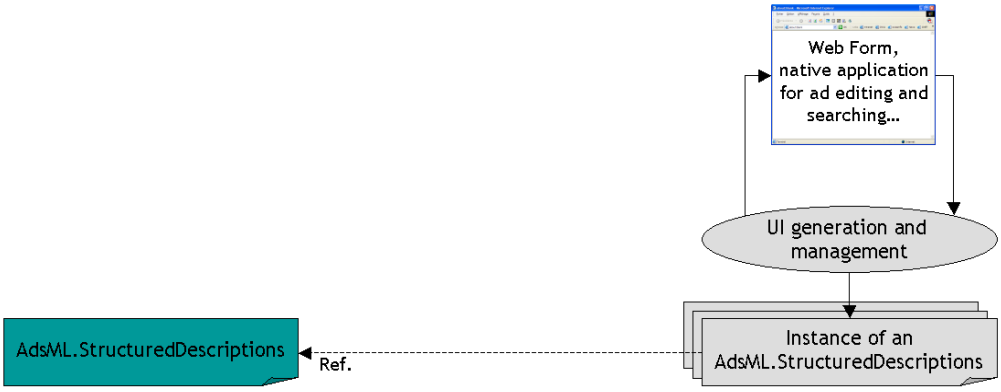
Note that more than one spreadsheet can be defined, if more than one type of advertising information is going to be exchanged between the systems and it is considered appropriate to divide the data definitions into multiple rule sets. Alternatively, a single heterogeneous rule set could be created that contains many different types of information, each on its own page.

5.1.2 Developer's view



Based on the definitions in the AdsML.AdObjectDefinitions instance, an application developer can implement a generic or a specific UI for ad editing, searching and displaying.

5.1.3 Advertiser's view, Consumer's view

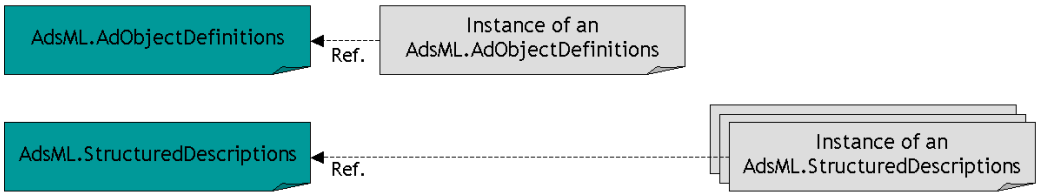


The Advertiser is able to create ads. The Consumer is able to browse or query online ads in order to shop for goods or services.

5.1.4 End user's view

When placing an ad, the advertiser will create/edit one AdsML.StructuredDescriptions instance per ad object, using attributes defined in the ad object definition. When looking at an ad, the end user (consumer) would search and browse online ads using those same attributes.

5.1.5 Content syndication view

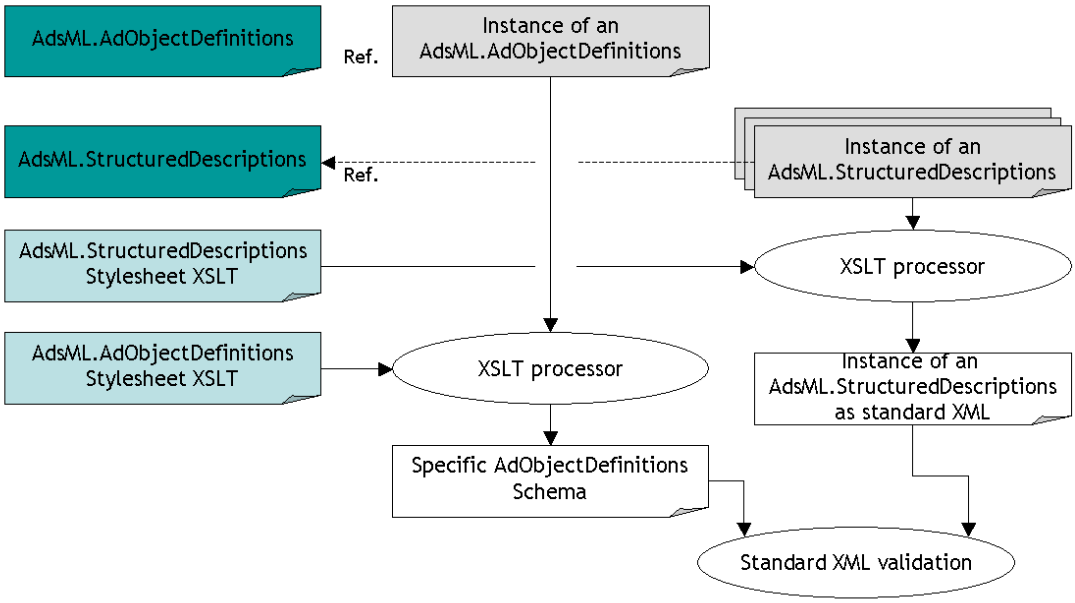


Two sites that agree to exchange content, should:

- Exchange AdsML.AdObjectDefinitions instances for each of the planned advertising categories. This is their formal confirmation and definition of the information formats they will use.
- Exchange the agreed ad instances in the form of AdsML.StructuredDescriptions files that conform to the Ad Object Definitions they previously exchanged.

Note that a single Structured Description message can contain information that is defined in several Ad Object Definitions files.

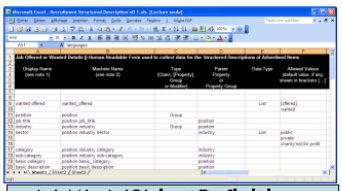
5.1.6 Content syndication validation view

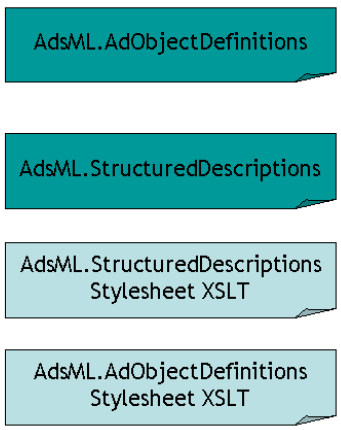


In addition to the previous syndication scenario, two sites doing syndication could also validate their content using AdsML-provided XSLT, by converting the structured descriptions instances into an XML format that can be validated using a validating parser, and converting the object definitions files into a schema that can perform the necessary validations.

5.2 Artifact list

As can be seen in the previous views, the AdsML Structured Descriptions standard defines or provides the following artifacts:





- AdsML Structured Descriptions **Rule Set** data entry interface – for example, the spreadsheet described above.
- AdsML **Ad Object Definitions Converter** – a tool for converting the rules defined in an AdsML Rule Set spreadsheet into the Ad Object Definitions format.
- AdsML **Ad Object Definitions** (AdsML.AdObjectDefinitions) – an XML format for representing a Rule Set. This is a compact XML structure that can express all of the information content of a Rule Set spreadsheet, plus a little more. The standard provides a schema for this format, and in practice, each rule set will be stored as an XML instance conforming to this schema.
    - NOTE: The extra capabilities that can be expressed in this format but not in the spreadsheet version of the rule set are relatively minor and technical, and will not be required in most markets.
    - NOTE: The format is optimized to express the rule set in a compact form that you can "see at a glance".
- AdsML **Structured Descriptions** (AdsML.StructuredDescriptions) – an XML format for representing the objects in a specific advertisement according to the rules specified in one or more AdsML Ad Object Definitions. Again, this is a compact XML structure that is optimized for processing by generic tools. The standard provides a schema for this format, and in practice, each set of structured descriptions about a given advertisement will be conveyed in an XML instance that conforms to this schema.
- AdsML **Structured Descriptions Validation Stylesheet** (AdsML.StructuredDescriptions Stylesheet XSLT) – converts an AdsML.StructuredDescriptions instance into a format that can more easily be validated by an appropriate XML Schema.
- AdsML **Ad Object Definitions Validation Stylesheet** (AdsML.AdObjectDefinitions Stylesheet XSLT) – converts an AdsML.ObjectDefinitions rule set into an XML Schema that can be used to validate an instance message that has been created using the AdsML.StructuredDescriptions Stylesheet mentioned above.

6 Generating the XML definition of a rule set from the spreadsheet version

6.1 Overview

Any rule set that has been recorded using the AdsML-defined structured descriptions spreadsheet interface can be converted into a formal Ad Objects Definitions XML document without loss of information. In general, the process of mapping the information into from the spreadsheet into the XML format is straightforward, once you are familiar with the XML structures:

- Each page of the spreadsheet (other than the initial "header" page) becomes an <AdObjectDefinition> element
- Each row of data on that page becomes either an <ObjectProperty>, <ObjectPropertyGroup> or <ObjectPropertyModifier> element
- Most of the columns of data in the spreadsheet become attributes of the XML element in question

- However, if a set of more than one "Allowed Values" have been defined for a given property or modifier, they are placed in a <ValueTable> element at the bottom of the XML structure, which is then referenced from the [attribute name] attribute of the object or modifier in question.

The following sections describe exceptions to the normal practice.

6.2 Property re-use

The spreadsheet interface allows for the re-use of property definitions, by creating an empty property group and specifying as its "data type" the name of another page of the spreadsheet where the relevant property definitions can be found.

However, the design of the Ad Objects Definition XML format does not permit this kind of property re-use. Therefore, when generating the XML version of a rule set that contains re-used properties, in each such case the definitions of the re-used properties must be copied into the place where they were referenced, replacing the external reference.

For example, the AdsML model vocabulary for Miscellaneous Goods contains several property groups that re-use the "telcom_number" structure, which is defined on another page:

	Display Name	Type (Object, [Property], Group or Modifier)	Data Type
48	telephone	group	telcom_number
49	mobile	group	mobile_telcom_numbe
50	fax	group	telcom_number
51	pager	group	telcom_number
52	instant messaging	group	Email address

Suppose that "telcom_number" contained only two properties: "area_code" and "number". In that case, when the XML for this section was generated, its structure (though not its syntax) would look something like this:

```
When_available (group)
  telephone (group)
    area_code (property)
    number (property
  …
  fax (group)
    area_code (property)
    number (property)
  pager
    area_code (property)
    number (property)
  etc.
```

As you can see, each property group whose data type was "telcom_number" has now been populated with the property definitions that were found on the referenced page of the spreadsheet. In the XML format, property groups are not allowed to have data types, so the re-use references have disappeared entirely.

6.3 Machine names for nested items

In the XML version of a rule set, the machine name for a given property, group or modifier must be unique *within the entire object definition* of which it is a part. No two properties, property groups or even modifiers within an object definition can share the same machine name.

In the spreadsheet version of the rule set, however, the only requirement is that each machine name must be unique *among its immediate siblings*. For example, all of the properties within a given property group must have unique machine names, and all of the modifiers of a given property must have unique machine names, but the same machine name might be used in two different property groups or for two different modifiers. (In fact this will often be the case, since some names such as "name", "size", "currency" and "unit of measure" may appear repeatedly in the rule set.)

In order to ensure that the machine names in the XML version of a rule set are appropriately unique, we recommend the following two principles be applied when converting spreadsheet names into the XML format:

- The machine name for a modifier should always begins with the name of its parent property using x.y notation. For example, the machine name of the "currency" modifier in a spreadsheet might be "currency", but in the XML version of that rule set it would be expanded to include the name of its parent property, e.g. "price.currency".
- The machine name for a property that is part of a property group always begins with the name of the parent property group. For example, suppose a property group named "Prices" contains a property called "Asking_price". In the spreadsheet, the property's machine name might be just "asking_price", but in the XML version it would have to be expanded to include the parent property group's name, e.g. "prices.asking_price".

These two naming conventions are additive, so that the "currency" modifier of the "asking_price" property in the "prices" property group would be named "prices.asking_price.currency".

7 Appendix A: Excerpts from the formal Requirements

This section contains excerpts for the AdsML Phase II Requirements document, which further amplify its vision and deliverables.

7.1 Vision

To create a globally applicable XML message structure and supporting tools (including controlled vocabularies) that are capable of describing the objects offered for sale in any kind of advertisement (including but not limited to Classified ads) in any media, in order to support:

- the process by which two or more trading partners agree on the structure and contents of the information they will exchange about advertised items,
- the controlled capture of this information,
- the automated classification, syndication and aggregation of advertisements by multiple publishers,
- and the searching and querying of the contents of those ads by potential consumers, both within and across international boundaries.

This message structure will provide a replacement for and superset of the equivalent functionality contained in the CREST 2.0 standard, but using an approach that meets the needs of the global market.

7.2 Deliverables

The deliverables of this activity **SHALL** include:

i. A documented approach to the design of XML message formats and their constraining schemas that will support transmission of structured descriptions of the properties of advertised items and related information in any type of advertisement
ii. A human-readable Form (e.g. a spreadsheet or document template) that users can populate with the properties of the specific types of advertised items that will be exchanged between their systems
iii. Rules for translating a populated form into a machine-processable rule set that defines and can be used to validate an XML message format suitable for conveying that information.
iv. Rules and techniques for using the governing rule set to validate the structure and contents of messages conforming to the XML message format.
v. Instances of combinations of the human-readable form, rule set and XML message format that have been populated with detailed sample vocabularies for at least the following classes of advertised items:
    a. Transportation
    b. Real Estate
    c. Recruitment
    d. Travel
vi. Instances of combinations of the human-readable form, rule set and XML message format that have been populated with detailed sample vocabularies for common properties of an Advertiser (for example, the advertiser's name and contact details).
vii. User guide:
    a. Business level user guide, including
        i. Overview of the business purpose and usage scenarios for the components of this standard
        ii. Business-level instructions for the use of the human-readable form
    b. Technical level user guide, including
        i. instructions for the population of the human-readable form, including naming conventions for the classes, properties and modifiers defined on the form
        ii. instructions for the technical processes involved in converting one or more human-readable forms into interoperable message formats, rule sets and validation routines that can convey and process the information defined in the forms

8 Appendix B: Acknowledgement for contributions to this document

Acknowledgement and thanks for contributions to this document are also due to:

- Members of the AdsML Technical Working Group

