STAR®

Standards for Technology in Automotive Retail

STAR Naming and Design Rules

Version 1.0

March 2007

Revision History

Revision	Date	Version
Initial Version	March 13, 2007	1.0

Table of Contents

STAR Naming and Design Rules

1. INTRODUCTION

1.1 PURPOSE

The purpose of this document is to define the naming and design rules that STAR uses to create the BODs. Where necessary, STAR may clarify the implementation of a specific rule, where we may deviate from the written definition but not the intent. STAR will only clarify rules where it deviates from either OAGI or UNCEFACT.

1.2 OBJECTIVE

This document only describes the rules, and is provided for informational purposes only. STAR members are to not use this to modify or create their own BODs. Any changes that are needed should still be submitted to the STAR data architects for review and implementation.

1.3 SCOPE

This document describes the naming and design rules for STAR BODs. This document does not go into aspects that are needed for implementation of the STAR BODs. This information is covered in the companion document, *STAR BOD Implementation Guidelines*.

1.4 PREREQUISITES

It is recommended that the following documents be read before reading this documentation.

- OAGI 9 Naming and Design Rules version 0.7
- UNCEFACT Applied Technologies Group Naming and Design Rules for XML version 2.0

STAR's Naming and Design rules leverages these two documents. This document will at times refer back to specific rules within the ATG and OAGI documents, when necessary.

1.5 REFERENCES

STAR has migrated its repository to leverage OAGIS 9, and UNCEFACT Core Components technology. To this affect, this document will reference the naming and design rules from each of these organizations. The order of preference will be:

- STAR Naming and Design Rules
- Open Applications Group (OAGi) Naming and Design Rules
- UNCEFACT Applied Technologies Group (ATG) Naming and Design Rules

Where OAGi and ATG 2 rules differ, the preference will be to apply the OAGi rule. ATG 2 Rules are used instead of OAGi rules in rare occasions, but since OAGi enhances and is compatible with ATG the STAR implementation is still compliant with OAGIS 9.

1.6 DOCUMENTS

STAR's Naming and Design rules are based on the following published document versions.

- OAGi Naming and Design Rules Version 0.7
  - Available: http://www.oagi.org/download/oagis/loadfrmNDR9.htm
- UNCEFACT ATG 2 Naming and Design Rules Version 2.0
  - Available: http://www.unece.org/cefact/xml/xml_index.htm
- UNCEFACT Core Component Library Version 1.0 February 2006
- UNCEFACT Core Components Technical Specification 2.01

STAR also maintains a version of these documents on it's website.

1.7 CONTACTS

Please send any comments and suggestions pertaining to this document to David Carver at dcarver@starstandard.org or Michelle Vidanes at mvidanes@starstandard.org.

2. BOD BASICS

2.1 BOD DEFINITION

BODs (Business Object Documents) are the business messages or business documents that exchange data between applications and components between companies. The BOD provides a common horizontal message architecture across multiple industries, with Automotive retail being just one such vertical industry. The BOD is a business message that is exchanged between application components and is

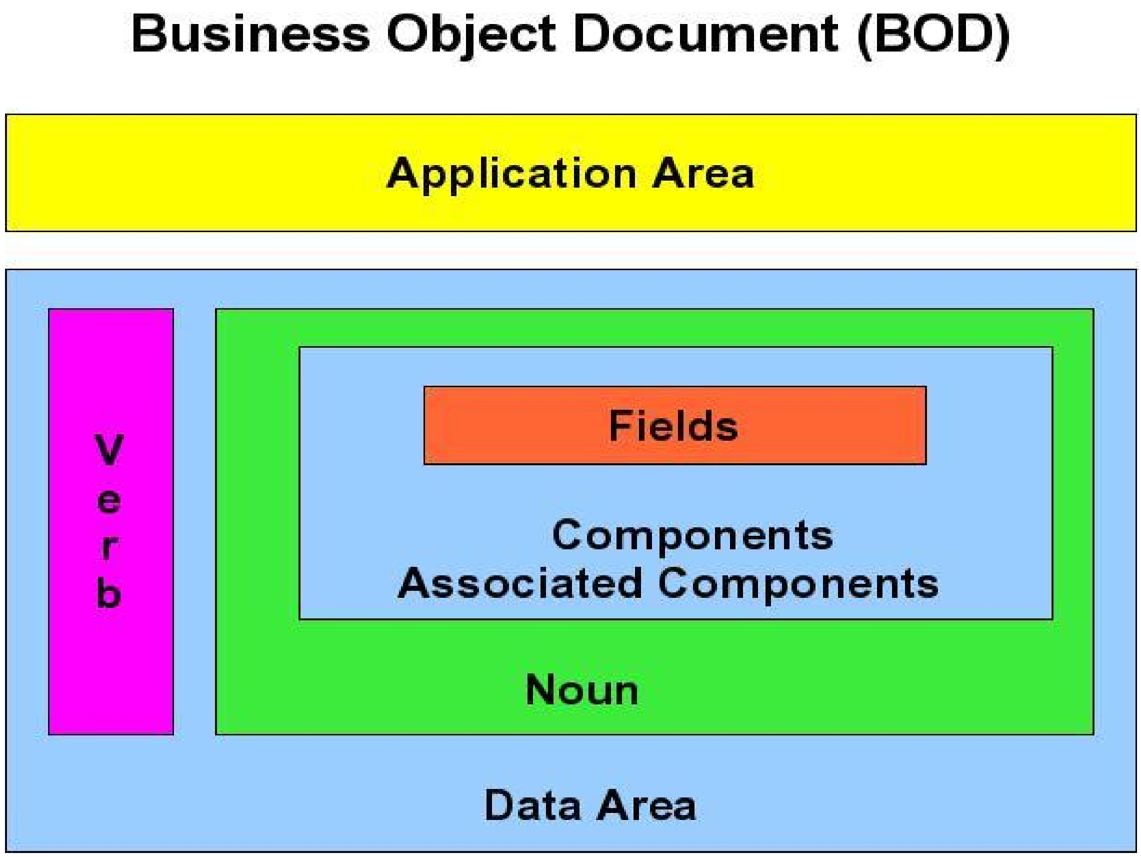

Figure 1: Business Object Document

independent of the communication mechanism. BODs can be used with various types of transport protocols such as SOAP and EBXML. However, the BOD Message Architecture is independent of the communication mechanism.

As depicted in Figure 1, All BODs have two high-level parts:

- An Application Area
- A Data Area

These form the basis for all BODs.

2.2 APPLICATION AREA

The BOD Application Area communicates information that can be used by the infrastructure to communicate the message. This includes information on the Sender of the BOD, date and time the BOD was created and information on the Destination of the BOD.

2.3 DATA AREA

The BODs Business Data Area includes a definition of the data, making it a self-describing message format. It can contain one or more occurrences of the data values. The data area is made up of the Verb and Noun.

2.3.1 Verb

The Verb identifies the action being performed on the specific Noun of the BOD. (i.e., Dealer initiates with a **Get**PartsOrder, OEM responds with an **Show**PartsOrder)

There is a predefined list of verbs (provided by OAGi) that STAR uses. This is information that is derived directly from OAGi and is applied to the STAR Nouns. Verbs constrain, add behavior and implied process to the Noun. Refer to the companion document, *STAR BOD Implementation Reference* for more information regarding implementation of the OAGi verbs.

2.3.2 Noun

The Noun identifies the business specific data that is being communicated (i.e., Parts Order, Repair Order, Credit Application, etc). Noun behavior is affected by the associated Verb. The Noun is where the individual data requirements are defined. The Noun consists of Components, and Fields.

2.3.2.1 Components

Components are the building blocks of a Noun. The components allow STAR to organize the data requirements in logical groupings. These groups could be vehicle related information, financing information, or information common to the various parties in a business transaction.

STAR's objective when designing components is to develop as many reusable components as possible. STAR will also leverage existing OAGi components where there structure will meet STAR members needs. In addition, STAR will also build components based on the information provided by the UNCEFACT Core Components Library. By leveraging both OAGI and UNCEFACT Core Components, it helps provide a more inter operable set of BODs and business semantics.

2.3.2.2 Fields

STAR and OAGI define a set of reusable data items, known as fields. These provide the basic business information for a particular component. Field information includes such items as Company Name, Part Identification, Document Date Time, etc. STAR has defined most fields as global references. In this way a field can be defined one time, and then reused in other components. This helps provide a common definition and usage for a particular field.

According to *ATG 2 NDR Rule [R 105]*, states that fields should be limited in scope to the component that they are defined (ATG 44). OAGI relaxes this rule, and allows for the use of references in the components to common field usage. STAR follows the OAGI rule for fields, allowing for either local or global definition.

3. SCHEMA

The XML Schema language is the Open Applications Group's Integration Specifications (OAGIS) recommended method for creating schemas. XML Schemas provide a rich syntax for expressing metadata. Some of its features include:

Structures are defined that allow the definition of relational (keyed) data, and object-oriented (type inheritance) data.

- Elements and attributes for structural constraints
- Schema uses the same format as XML document.
- Schema uses XML namespace support for extensibility. Namespaces enable developers to avoid naming collisions by assigning element and attribute names. An XML name is prefaced by characters defining the namespace it belongs to.

3.1 STAR SCHEMA STRATEGY

STAR uses a global strategy to develop the schema:

- Schema are created in STAR namespace referencing OAGI and W3C namespaces for types, nouns and verbs.
- STAR uses Global field definitions instead of local.

OAGi allows for the definition and construction of a BOD using either locally scoped or global elements. STAR has chosen the global element approach to allow for consistent reuse within it's schemas. It is STAR's intent to have common structures where possible for similar information. If a Party is defined in one way, it is easier to implement across the BODs instead of having 79 different ways a party could be defned.

3.2 DATA DICTIONARY

The STAR XML data dictionary is the Fields file. Most of the fields have been defined as optional rather than required. Only when all members of each specific work group agreed that a field should be required could we define it as required.

3.3 OAGI USERAREA

STAR does **not** support the use of the OAGIS “UserArea” which is added to all OAGIS components. Use of the “UserArea” would make each implementation unique and would be contrary to supporting common implementations across STAR members.

STAR breaks data into reusable components wherever possible. STAR develops small components that can be combined as required. This is the same approach that OAGI uses.

3.4 USE OF SUBSTITUTION GROUPS AND XSI:TYPE

In prior versions of OAGi and STAR BODs, the use of substitution groups was used to allow other components to be replaced or used instead of the ones that had been defined. This lead to some interoperability issues, and deviated from the intent of providing a common standard. Also, to this affect, many implementations specified the xsi:type attribute in the XML. This is itself a form of substitution, and is no longer allowed. STAR supports the following ATG 2:

- *ATG [R202*] - The xsi:type attribute MUST NOT be used. (ATG 76)
- *ATG [R65]* – Substitution groups MUST NOT be used. (ATG 103).

4. NAMING AND DESIGN RULES

STAR supports and follows the naming and design rules outlined by the OAGi and ATG documents. STAR's interpretation of the rule documents are as follows:

- OAGI supports and follows all rules specified within ATG unless specifically saying otherwise.
- STAR is allowed to use the ATG Rule if it is preferential to the OAGi rule.

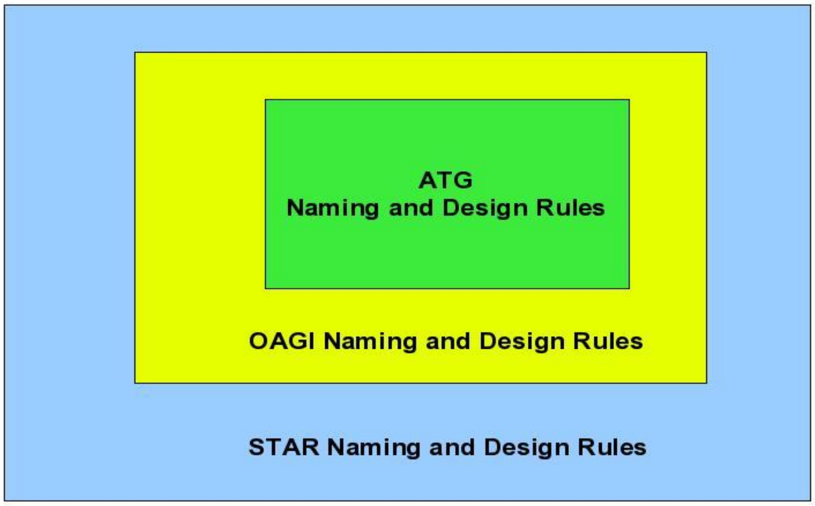

Figure 2: Naming and Design Rules

The sections that follow will only cover items where STAR deviates from the OAGi Rule.

4.1 NAMESPACES

OAGI defines a format for the use of Namespaces for the BOD methodology. This is defined in OAGI [R 54], which specifies the namespace should take the following format:

<URL>/<standard>/<major>/[<overlayname>|<substandardname>]/[<overlay major>|<substandardmajor>]]

An example, OAGI namespace would be:

http://www.openapplications.org/OAGI/9

STAR adopts in the intent of this rule, but specifies the namespace for all STAR BODs as follows:

http://www.starstandard.org/STAR/5

STAR does not adopt OAGi rule [R 56], which states that "extensions to OAGi BODs, must fall under the OAGi namespace". STAR rarely uses OAGi BODs, and but does leverage the Components in the BODs. STAR maintains a separate repository from OAGi, and uses OAGi primarily as a methodology and a library.

4.2 SCHEMA LOCATION

STAR adopts OAGI Rules [R 58, R 59, R 60, R 61, and R 62], with the following clarifications. Even though an XML Instance can contain a schemaLocation attribute, it is recommended as a best practice not to rely on this for schema identification or location. Implementations are free to ignore this. It is also recommended that the namespace, and root element be used to identify the repository a particular BOD should validate against.

If schemaLocation is used it should only contain the BOD name as a hint on what schema to use when validating the BOD.

4.3 EMPTY FIELDS

STAR does not support fields to be empty. In prior versions of the schemas, NULL values could be transmitted. However, this did not comply with the STAR data compliance guidelines which forbid transmitting NULLs. To this affect, STAR has implemented within the schema the requirement that all string based fields have a minimum length of at least 1. This enforces within the schema this data compliance rule. This is a rule that is also stated by OAGi and ATG naming and design rules, but isn't explicitly enforced within the schema that is generated.

4.4 USE OF UNCEFACT CORE COMPONENTS LIBRARY

STAR makes use of the OAGIs Core Components as well as other Core Components defined by UNCEFACT. Where STAR has leveraged a UNCEFACT Core Component, the complexType name will have the words ABIE within it. This allows for the clear identification of a UNCEFACT Core Component. This naming will only appear in the complexType name, and not the element name itself. This allows STAR to change the complexType name at a later date to align with OAGI and UNCEFACT naming and design rules. This will happen as OAGi and UNCEFACT NDRs align, and as more implementations of the UNCEFACT Core Component library appear.

STAR will from time to time re-evaluate the components it uses, and may at a later date use a newer library than what has been stated in this document. Backwards compatibility where possible will be maintained.

5. WORKS CITED

Applied Technologies Group, *XML Naming and Design Rules*, 16 Feb. 2006. UNCEFACT. 13 Mar. 2007 http://www.cen.eu/uncefactforum/ATG/atg_news_download.htm

Open Applications Group, *OAGIS 9.0 Naming and Design Rules Standard*, 30 Sep. 2005. OAGi. 13 Mar. 2007