Zaandijk

ICT Strategy
ICT Architecture
Software Solutions

Digitect B.V.
Postbus 30
1540 AA Koog aan de Zaan
The Netherlands

E-mail   mail@digitect.nl
WWW   www.digitect.nl

KvK Amsterdam 34161245

ISO Audit Data Collection Standard Modeled

Version 1.0

Recommendation to ISO TC 295

Authors

Frans van Basten
Senior Consultant

Jan Vrijenhoek
XML/EDI Consultant

2021-04-14
ADCS, Model

Zaandijk

ICT Strategy
ICT Architecture
Software Solutions

Digitect B.V.
Postbus 30
1540 AA Koog aan de Zaan
The Netherlands

E-mail   mail@digitect.nl
WWW   www.digitect.nl

KvK Amsterdam 34161245

2021-04-14
ADCS, Model

Content **Page**

1. INTRODUCTION....................2

2. ADCS MODELED....................3

2.1. Data modeling....................3
2.1.1. Business Chain Model....................4
2.1.2. Transaction Model....................7
2.1.3. Functional Message Model....................7

2.2. Modeling issues in relation to ADCS....................8
2.2.1. Modeling Business Segment information....................8
2.2.2. Modeling Account Segment information....................8
2.2.3. Order of fields....................9
2.2.4. Specifying Identifiers....................9
2.2.5. Naming conventions....................9
2.2.6. Additional modeling options to reduce number of elements....................10

1. INTRODUCTION

This document describes how the Audit Data Collection Standard (ADCS) has been modeled by Digitect and then used in the Dutch Proof Of Concept for development of the ADCS Customs Extension.

The created model is based on the International Organization for Standardization (ISO) specification: *"Audit Data Collection 20190711 FDIS PC 295 clean version"* by ISO/PC295/WG 1

2. ADCS MODELED

2.1. Data modeling

Digitect supports for many years the design and implementation of an information architecture, based on electronic messages or web services, by functional and technical specifications.

Digitect complies with the standard methodologies ISO 11179 (Specification and standardization of data elements by modeling) and ISO 17113 (Method for the development of electronic message specification).

The process of modeling consists of 3 steps;

- building an Object Model which is called the *Business Chain Model*,
- deriving a hierarchic *Transaction Model* from the *Business Chain Model* and,
- deriving one or more types of *Functional Message* models from a *Transaction Model*.

The Functional Message model is the basis to generate technical message specifications.

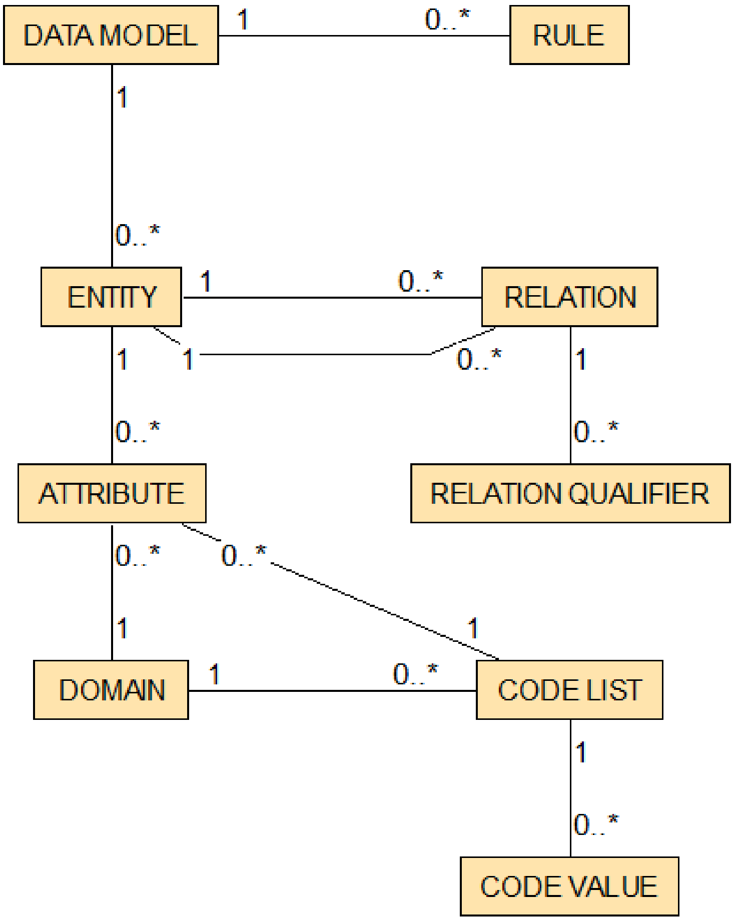
Four types of technical message specifications are supported:

- Flat file formats (CSV, TXT),
- XML Schema's,
- JSON Schema's,
- Edifact mappings.

Generating an XBRL Taxonomy from the model is in development.

Also a variety of RTF, HTML, XLSX documentation and sample messages with understandable data, generated automatically from functional specifications is supported.

2.1.1. Business Chain Model



Meta Model of the Business Chain Model

A Business Chain Model consists of the following elements:

- Data model
- Entities
- Attributes
- Domains
- Code lists
- Code values
- Relations between Entities
- Relation qualifiers
- Rules

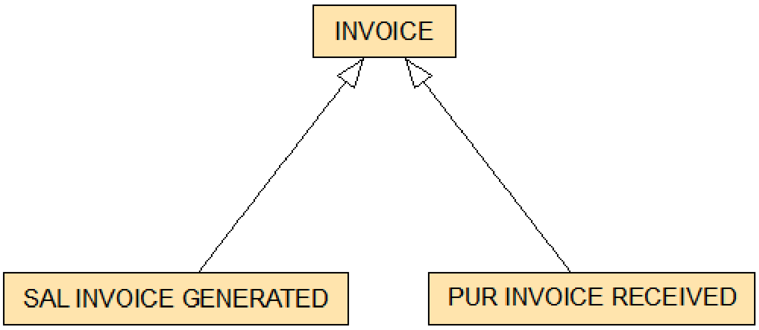
An Entity contains one or more Attributes (For instance entity *Address* contains attributes *Street Name*, *Postal Code* etc.)

Each attribute refers to a Domain. All Attributes of the same type refer to one and the same Domain, which contains the definitions for that type of attribute, which are format, data type, maximum length, pattern etc. and the domain refers to one or more code lists which are applicable for that type of attributes.

Two Entities can relate to each other in the model.

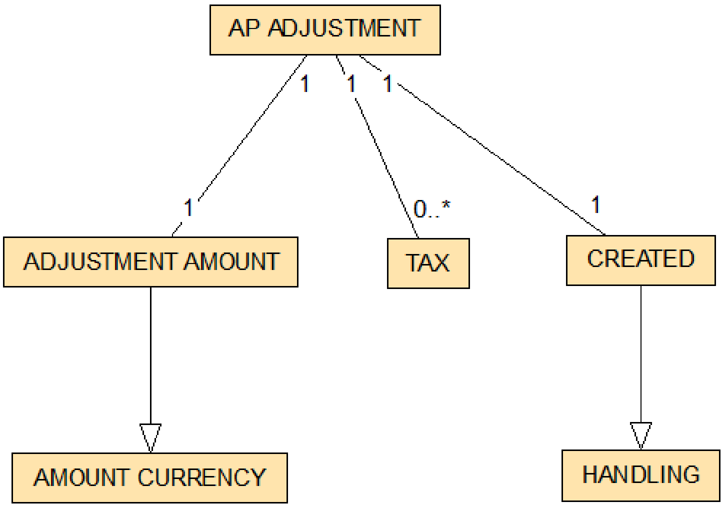
Examples of Entity - Entity relations used in the model are:

IS-A relation:



IS-A relation to INVOICE

1..1 and 1..n relations



1..1 and 1..n relations

Each table in ADCS, which contains data elements, is modeled as "parent" Entity with Attributes and 1 or more possible relations to other Entities, which contains reusable and repeatable sets of attributes (e.g. Tax).

In the ADCS model the following IS-A relations are created:

- Handling (Created, Approved, Last Modified, Posted)
- Contact (Primary Contact)
- Address (Physical Address, Billing Address)
- Contract (Purchase Contract, Sales Contract)
- Contract Details (Purchase Contract Details, Sales Contract Details)
- Order (Purchase Order, Sales Order)
- Order Details (Purchase Order Details, Sales Order Details)
- Invoice (Purchase Invoice, Sales Invoice)
- Invoice Details (Purchase Invoice Details, Sales Invoice Details)

In the ADSC model the following 1..1 relations from several parent Entities are created to:

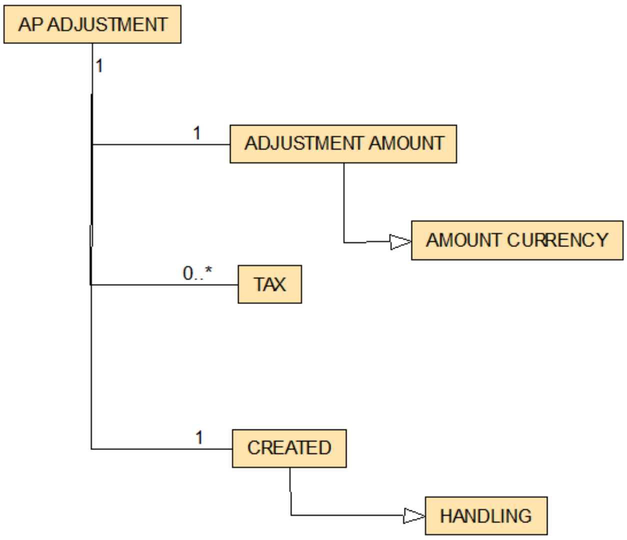
- Amount Currency
- Created
- Approved
- Last Modified
- Posted
- Accounting Segment

In the ADSC model the following 1..n relations from several parent Entities are created to:

- TAX

2.1.2. Transaction Model

In the Transaction model, all entities are placed in an hierarchic structure, for example:



Hierarchic structure of AP ADJUSTMENT

Each table in ADCS, which contains data elements, is placed on level 1 in the hierarchic transaction model. When such a level 1 Entity has relations to other Entities, these Entities are nested under the level 1 Entity.

2.1.3. Functional Message Model

From the transaction model, one or more Functional Message models can be derived.

In case of ADCS, one Functional Message model has been defined, with three technical representations, specified in:

1. Flat file message specification (CSV)
2. XML Schema specification (W3C)
3. JSON Schema (Version 0.6)

These three technical specifications will contain the same functional content specification but will differ on the technical level only.

2.2. Modeling issues in relation to ADCS

2.2.1. Modeling Business Segment information

Table No.	Field No.	Description	Solution	Status
All		Construction of variable field Business_Segment_X in ADCS can not be modeled in that way.	Change the name of field *Business_Segment_X* into *Business_Segment_Code*, which refers to *Business_Segment_Code* in table *BAS_Business_Segment* Add a field *Parent_Business_Segment_Code* to table *BAS_Business_Segment* Delete table *BAS_Business_Segment_Hierarchy.*	TC

2.2.2. Modeling Account Segment information

Table No.	Field No.	Description	Solution	Status
All		Construction of variable field Account_Segment_X in ADCS can not be modeled in that way.	Create a new BAS table, *BAS Account Segment*, with fields: *Account_Segment_Code* (key) *Account_Segment_Name* Create a Repeating Entity with fields: *Account_Segment_Code*, which refers to *Account_Segment_Code* in table *BAS_Account_Segment,* and field: *Account_Segment_Value.*	TC

2.2.3. Order of fields

Table No.	Field No.	Description	Solution	Status
All		Order of fields in the by ISO specified tables will differ from the order specified in the model, because of data modeling.	Accept the order specified in the model.	Done

2.2.4. Specifying Identifiers

Table No.	Field No.	Description	Solution	Status
All		Specifying Identifiers in ADCS is done in separate tables. In the model we use the Attribute properties: “Primary Key”, “Foreign Key”, “None”.	Set the Attribute property in the model to: “Primary Key”, “Foreign Key”, “None”. The name of the reference entity is captured in the attribute description.	Done

2.2.5. Naming conventions

Table No.	Field No.	Description	Solution	Status
All		Names of tables and names of fields in tables can differ from the names given in the model. Because of data modeling, some field names will not start with the table name.	Accept the names specified in the model.	Done
All		Names of tables are defined in “plural” form. E.g. SAL_Contracts. Best practice of modeling is to use singular names. In the model the entity name will be SAL_Contract.	Accept the entity names specified in the model.	Done

2.2.6. Additional modeling options to reduce number of elements

Table No.	Field No.	Description	Solution	Status
14,16		BAS_Customer and BAS_Supplier	Create Entity Customer/Supplier	ToDo
67,95		AR Adjustment / AP Adjustment	Create super type entity Adjustment with IS-A relations.	ToDo
69,97		AR Adjustment Details / AP Adjustment Details	Create super type entity Adjustment Details with IS-A relations.	ToDo
65,93		AR Cash Application / AP Cash Application	Create super entity Cash Application with IS-A relations.	ToDo
63,91		AR Cash Received / AP Payment Made	Create super entity Transaction with IS-A relations.	ToDo
140, 142, 144		PPE Addition / PPE Removal / PPE Change.	Create super entity PPE Transaction with IS-A relations.	ToDo