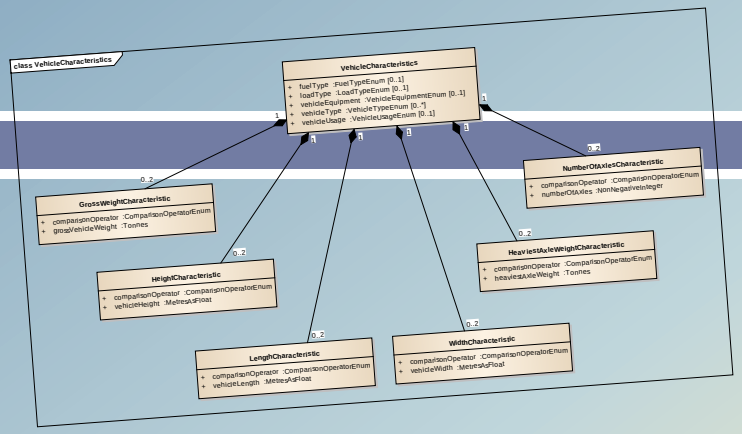
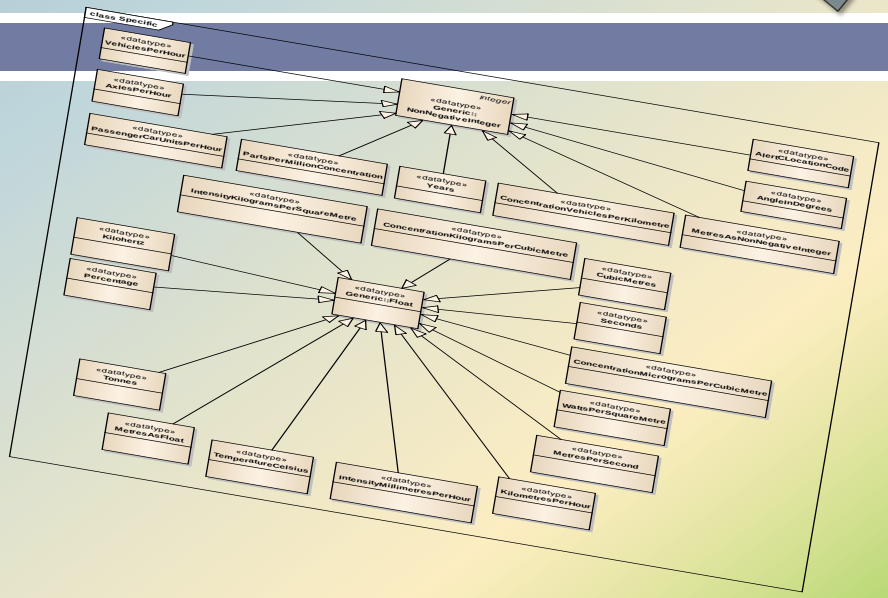
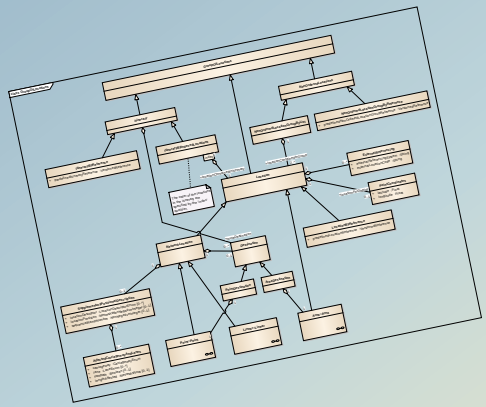






PROFILES AND EXTENSIONS

European Commission
Mobility & Transport

Jörg Freudenstein

About me

- **Jörg Freudenstein**
- Computer scientist, born in 1977
- Since 2005 project engineer at AlbrechtConsult (Aachen, Germany)
- Emphasis on software engineering / software processes, communication networks und distributed systems, software architectures as well as data modelling
- Specification of DATEX II-profiles for the German Mobility Data Market Place (MDM)
- Participation in the DATEX development and standardisation
- Editor of the DATEX II parking extension

Todays hands on workshops

Overview

- Basics — Now
- Tour through the DATEX Level A data model
- Creating of profiles
  - Data selection
  - Creation of subschemas
- Usage of extensions
  - Extension of the model with new content
- Creation of publications
- Perspective on current developments (e.g. intelligent truck parking)

What is DATEX II?

- An open, interoperable interface specification for the machine to machine exchange of dynamic road traffic and travel data between traffic centres and with third parties, including broadcasters and providers of commercial ITS services
- Consists of a set of artefacts:
  - Methodology (Meta data model ⊂ UML 1.4.2, rules) — partly
  - Data model (dynamic road traffic data)
  - Software to generate data schema (~ message syntax)
  - Documentation — partly
- Based on general accepted IT standards (UML, XML, http, ...)

Today

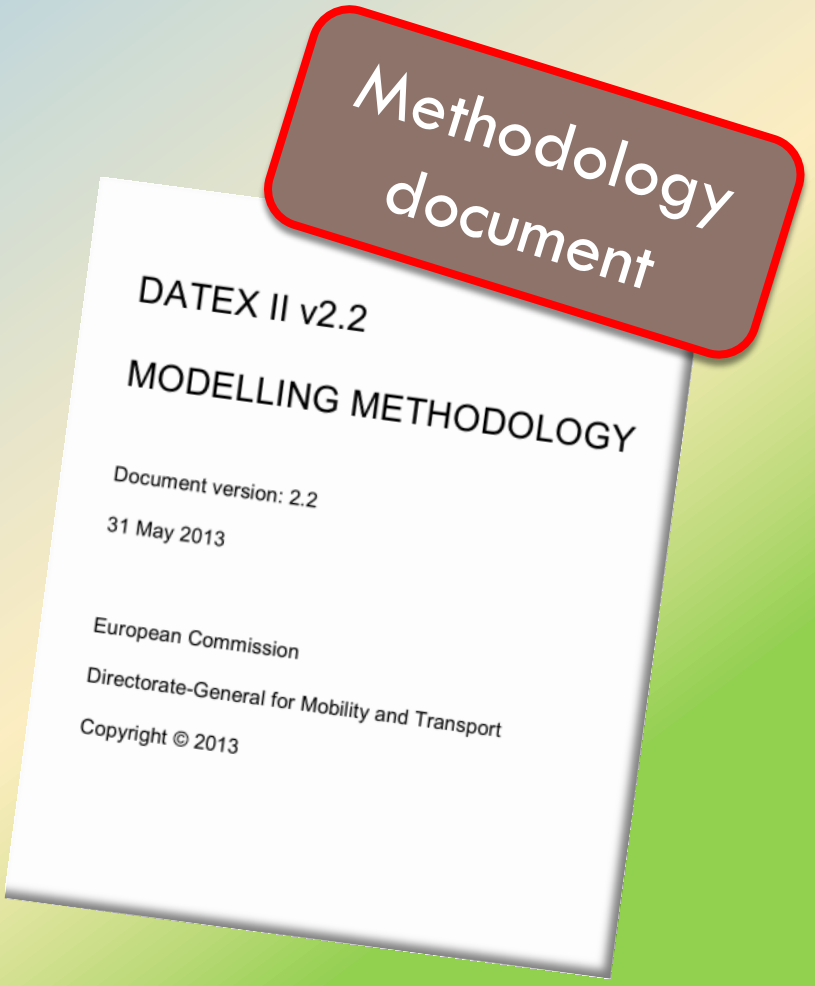
- Made available free of charge and without rights reserved by the DATEX user community (mainly road operators)
- → **standardized as CEN/TS 16157 Part 1 – Part 5** (as Technical Specification)

  (in a modified form, but content consistent)

DATEX Methodology

- Basis for the mapping of the data model in UML and for the creation of profiles and extensions
- Based on the division into three model levels:
  - M0: Data
  - M1: Data model
  - M2: Meta model

Where appropriate in this lecture, this methodology document is quoted.



Elements of DATEX II (v2.2)

Reference documents (normative)

- Data model (UML, as EAP-file or for web browser)
- Methodology (file „DATEX_II_Metodology.pdf“)
- XML schema
- XML-schema & software tool (Windows .NET) to generate a schema
- Exchange specification: text document (PDF) & WSDL-files

Additional documents (informative)

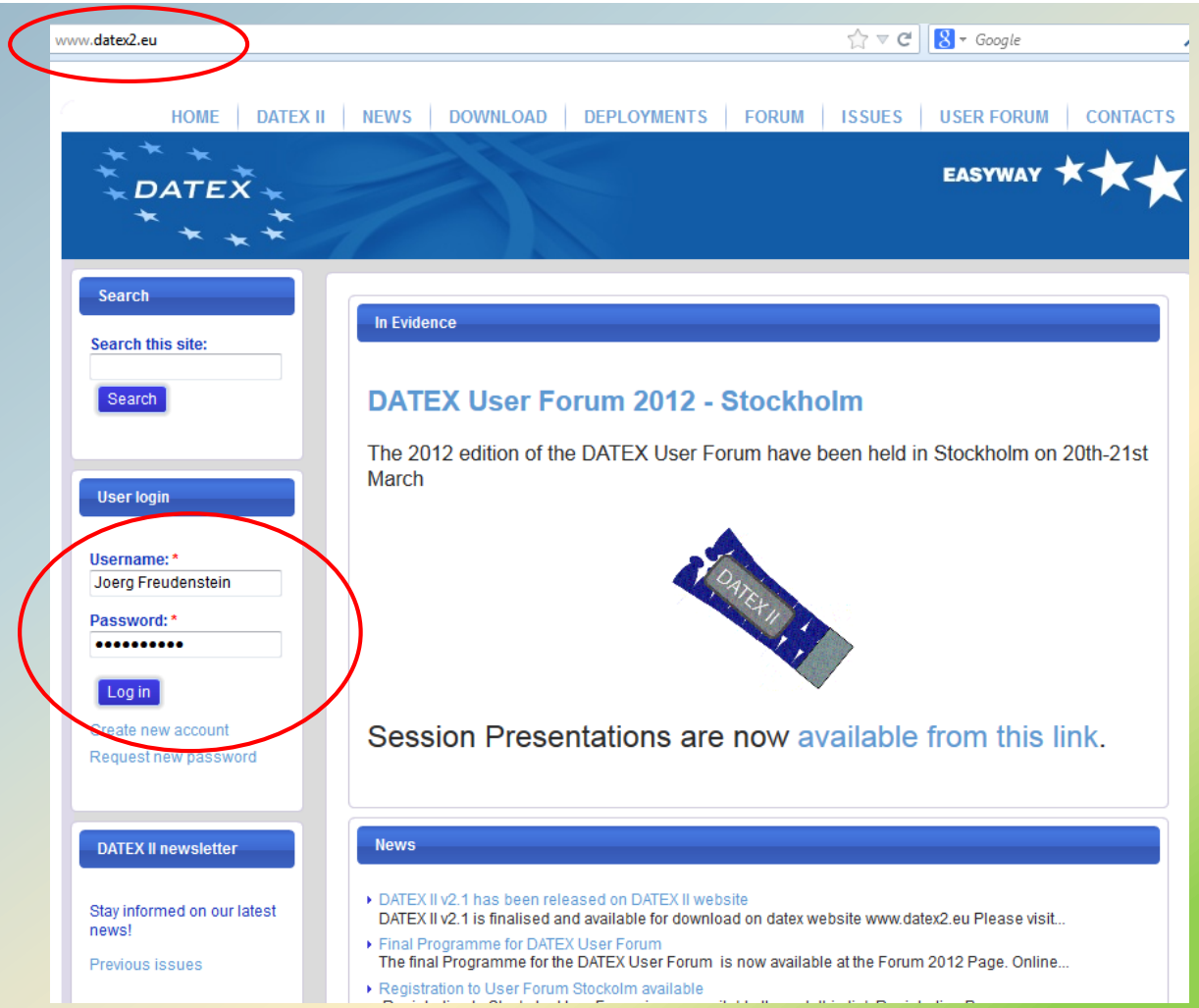
- Handbooks for users, for software development and the creation of extensions
- Documentation of the platform independent exchange (Exchange-PIM)
- Data Dictionary
- Handbook for the XML schema generation tool
- Example messages

DATEX II data model

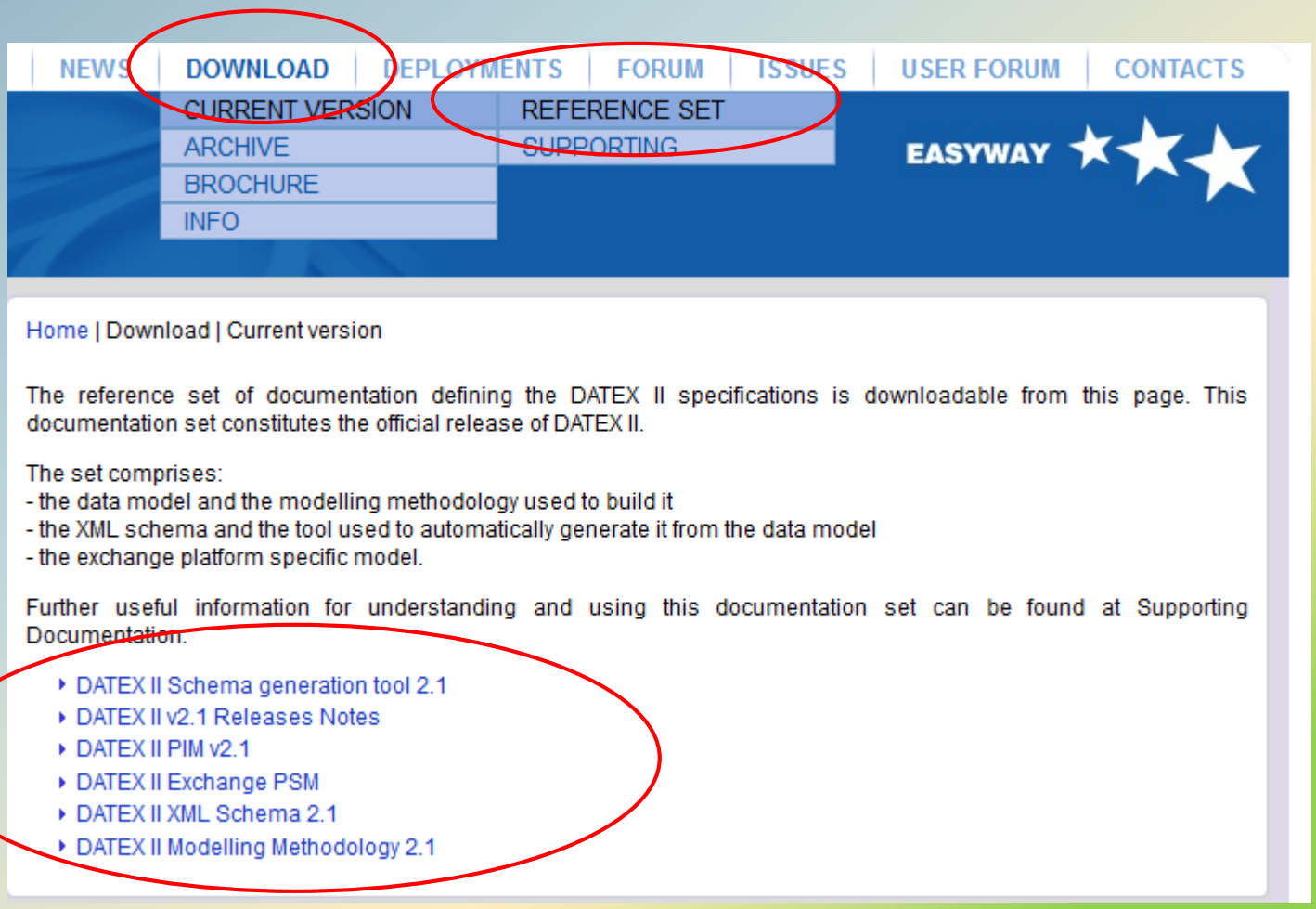
- The data model (and all other files mentioned) are available on the DATEX website.
- To get access on the current documents, a login (free registration) is necessary:

http://www.datex2.eu

Get access


www.datex2.eu
Google
HOME
DATEX II
NEWS
DOWNLOAD
DEPLOYMENTS
FORUM
ISSUES
USER FORUM
CONTACTS
DATEX
EASYWAY
Search
Search this site:
Search
User login
Username: *
Joerg Freudenstein
Password: *
Log in
Create new account
Request new password
DATEX II newsletter
Stay informed on our latest news!
Previous issues
In Evidence
DATEX User Forum 2012 - Stockholm
The 2012 edition of the DATEX User Forum have been held in Stockholm on 20th-21st March
Session Presentations are now available from this link.
News
DATEX II v2.1 has been released on DATEX II website
DATEX II v2.1 is finalised and available for download on datex website www.datex2.eu Please visit...
Final Programme for DATEX User Forum
The final Programme for the DATEX User Forum is now available at the Forum 2012 Page. Online...
Registration to User Forum Stockolm available

Get access


NEWS
DOWNLOAD
DEPLOYMENTS
FORUM
ISSUES
USER FORUM
CONTACTS
CURRENT VERSION
REFERENCE SET
ARCHIVE
SUPPORTING
BROCHURE
INFO
EASYWAY
Home | Download | Current version
The reference set of documentation defining the DATEX II specifications is downloadable from this page. This documentation set constitutes the official release of DATEX II.
The set comprises:
- the data model and the modelling methodology used to build it
- the XML schema and the tool used to automatically generate it from the data model
- the exchange platform specific model.
Further useful information for understanding and using this documentation set can be found at Supporting Documentation.
DATEX II Schema generation tool 2.1
DATEX II v2.1 Releases Notes
DATEX II PIM v2.1
DATEX II Exchange PSM
DATEX II XML Schema 2.1
DATEX II Modelling Methodology 2.1

DATEX II data model

- The data model (DATEX II PIM v2.2) is available as
  - Enterprise Architect format (.eap) (see before)
  - In HTML format for web browser, as offline ZIP or online browseable:

http://www.datex2.eu/datex-model/HTML.Version_2.2/index.htm

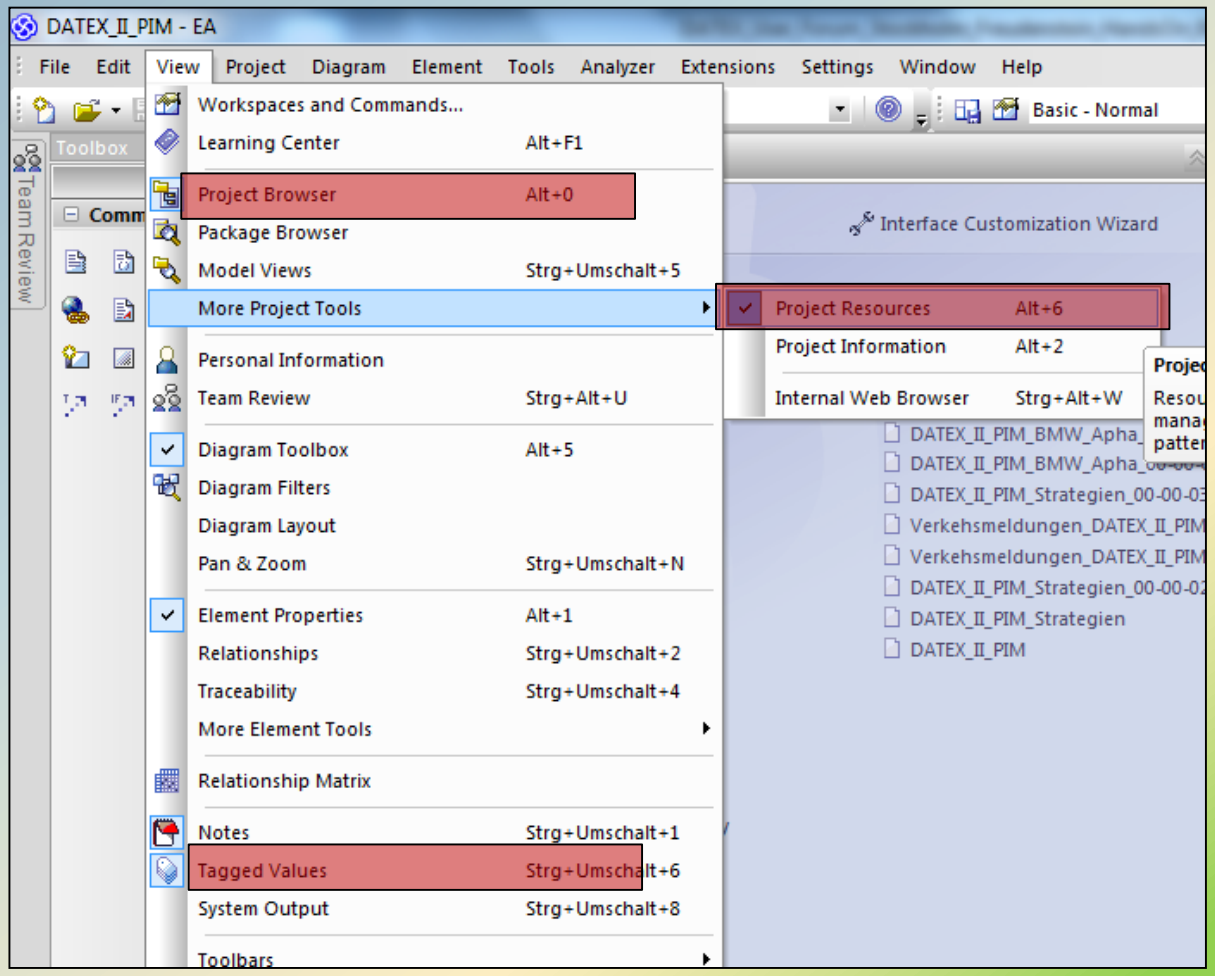
  - TamTam research developed an alternative online browser to search through the DATEX model:

http://datexbrowser.tamtamresearch.com

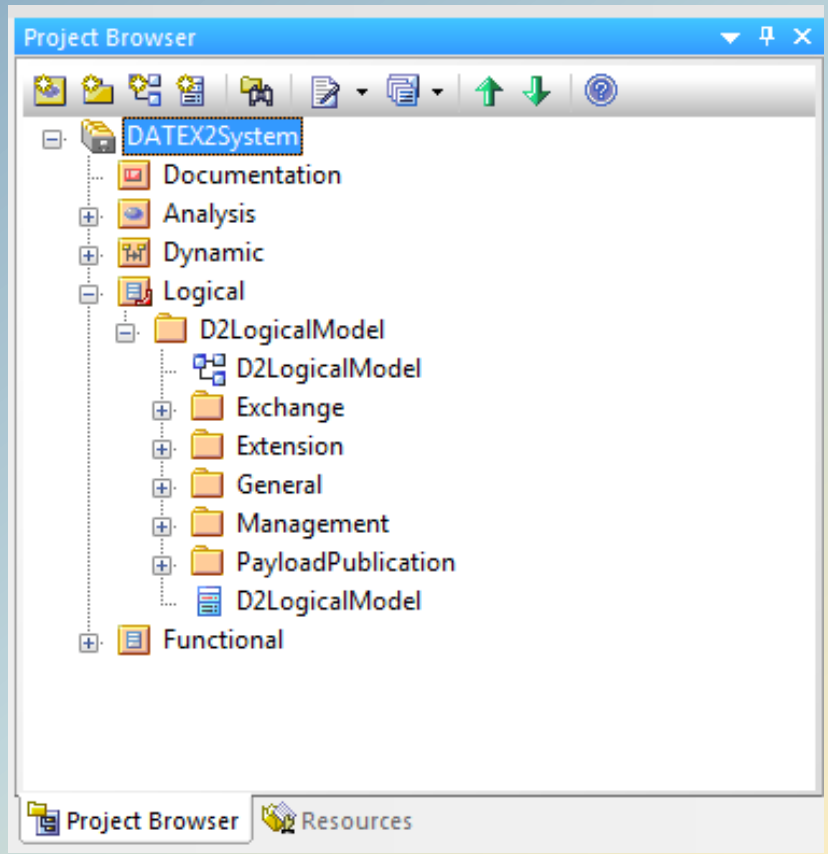
Enterprise Architect

- Inexpensive UML modelling tool used to view
- To obtain from Sparx Systems:
  http://www.sparxsystems.com/
- There is also a free version (viewing only) available!

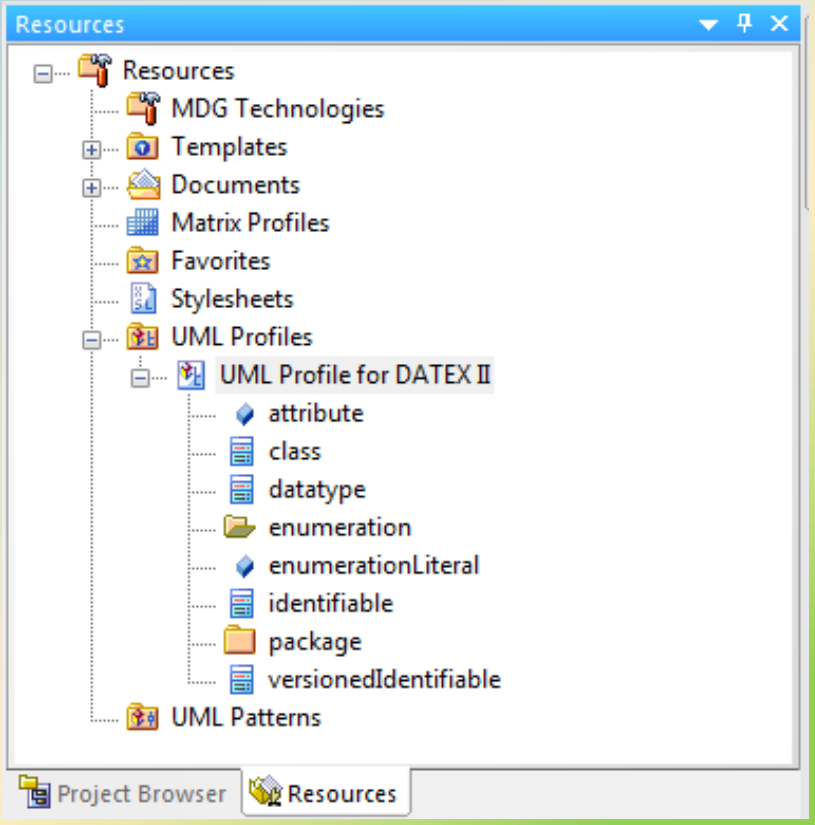
Settings in Enterprise Architect



Project Browser and Resources



Entry point for the model:
D2LogicalModel



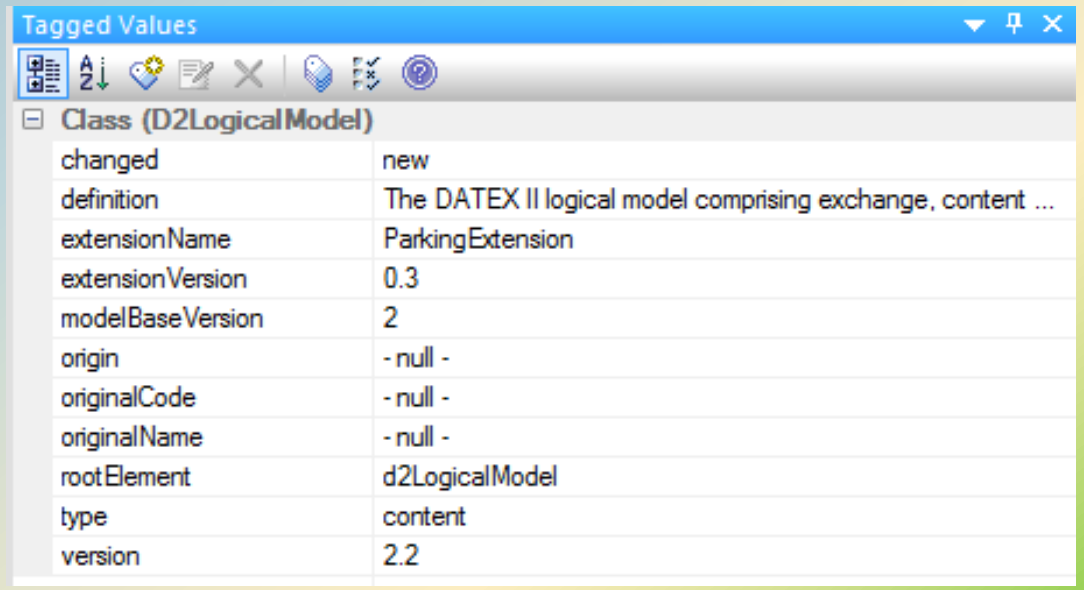
Special UML profile for DATEX for the generation of preconfigured UML elements

Tagged Values

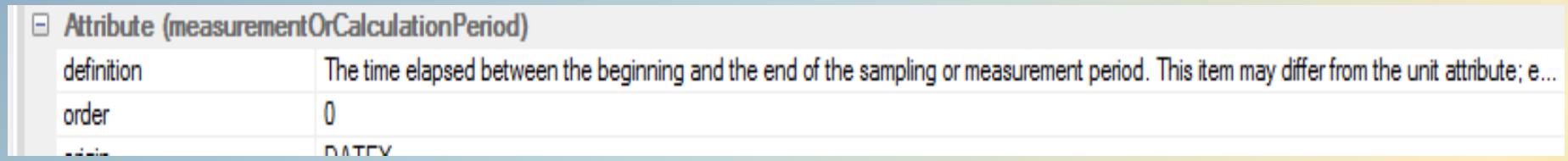
Tagged values offer user-defined additional information and consist of a ‘tag' and an associated ‘value‘ element.

Within the meaning of UML (1.4!) tagged values can be associated to any element in any quantity.

In terms of the DATEX methodology, the used tagged values are well defined, however.



Tagged Values (cont.)



Attribute (measurementOrCalculationPeriod)	
definition	The time elapsed between the beginning and the end of the sampling or measurement period. This item may differ from the unit attribute; e...
order	0
origin	DATEX

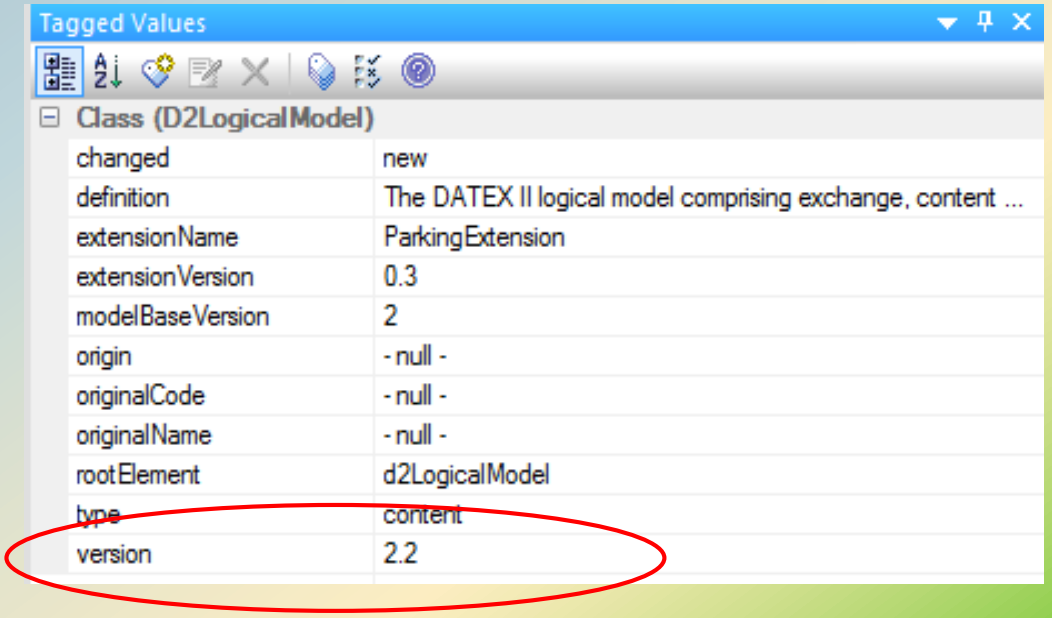
Two important DATEX tagged values:

- definition
  A definition for <u>each</u> component, attribute, literal and package.
  It becomes part of the schema and can be visualized by software tools.

- order
  An information for the relative position of this element within the schema.

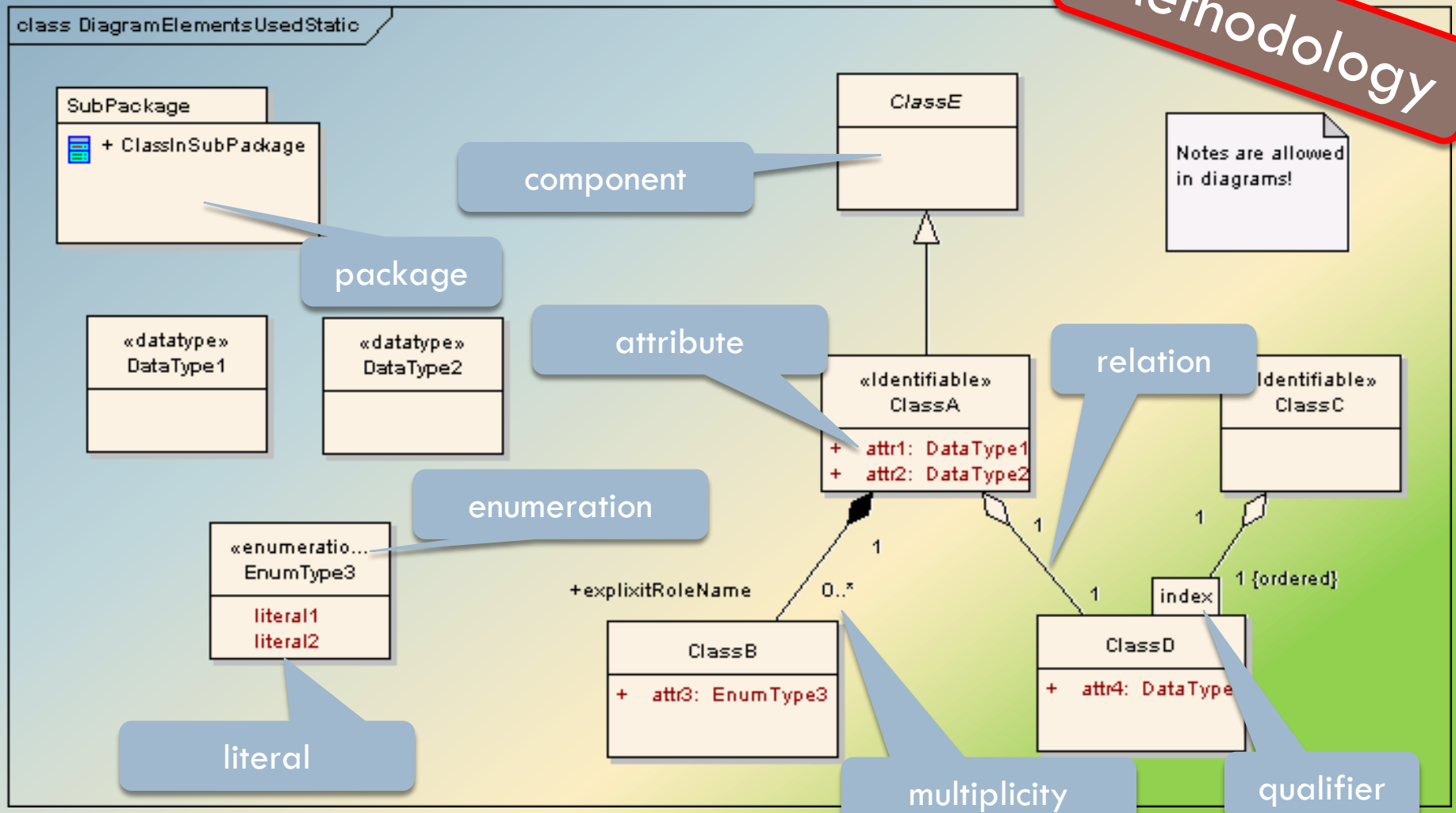
Check for the current version

Current version of DATEX II is 2.2.
(2.3. to come this year)
Older versions should not be used unless there is a need for it.
You can find the version information as a tagged value on the component ‚D2LogicalModel‘

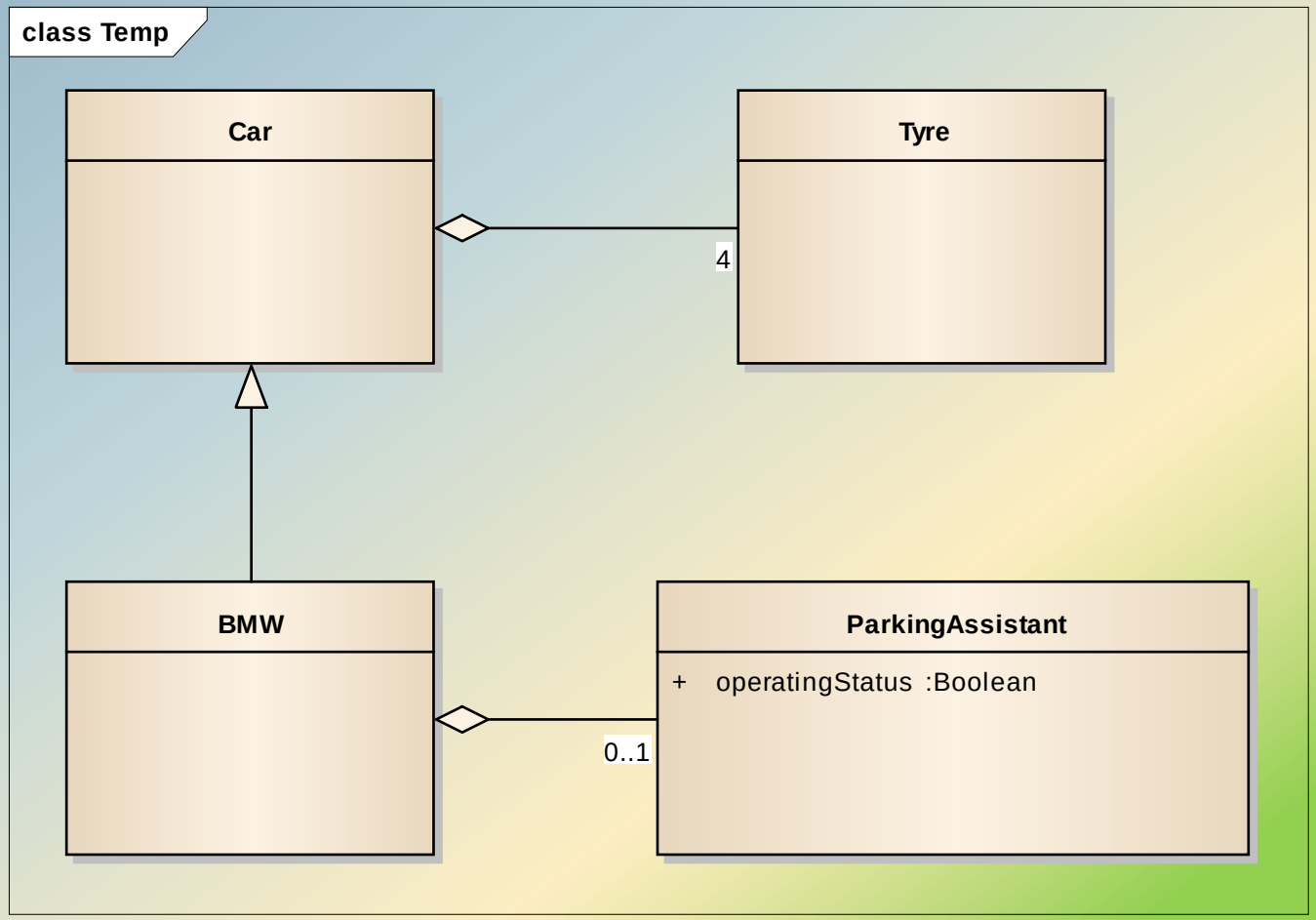


Used UML elements in DATEX II

Methodology



UML example

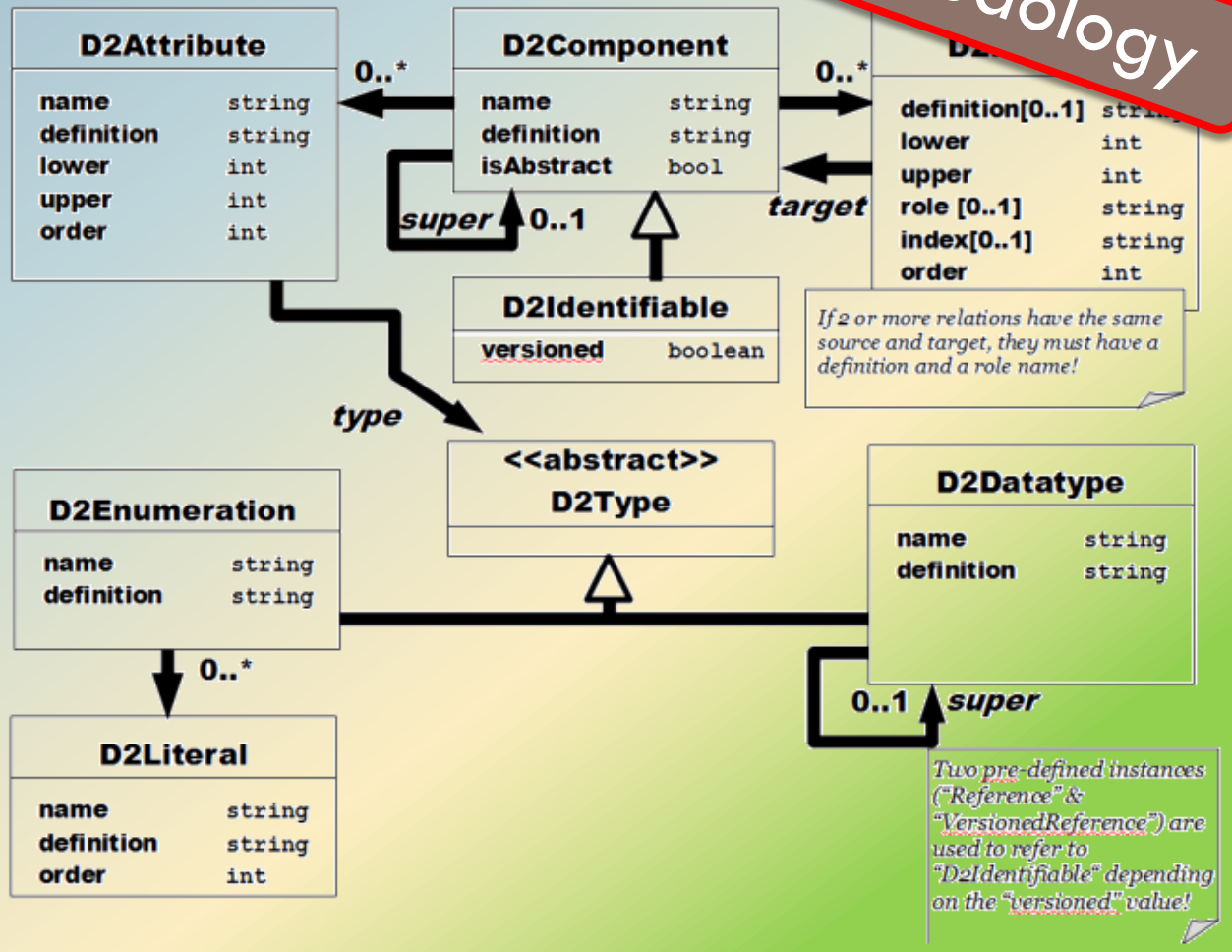

class Temp
Car
Tyre
4
BMW
ParkingAssistant
+ operatingStatus :Boolean
0..1

DATEX meta model

Methodology

An aside:

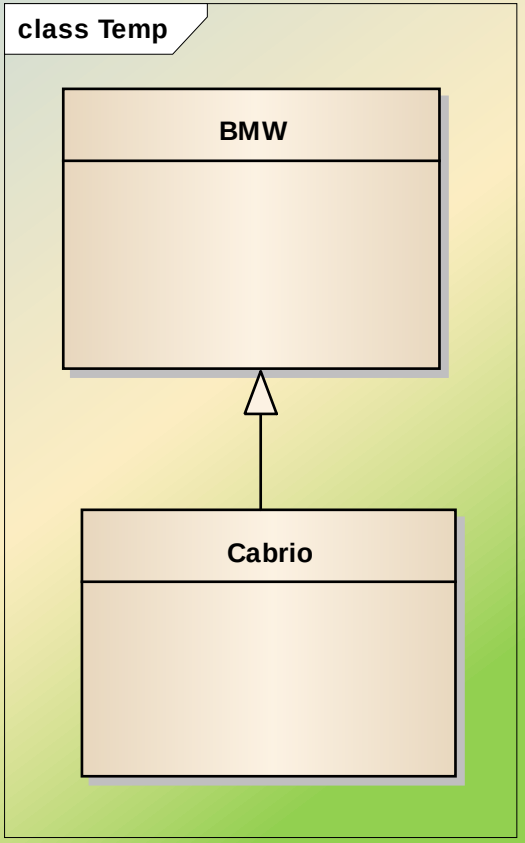
Attributes, relations etc. for DATEX are defined in a meta model with the means of UML



Different Views in Enterprise Architect

The data model can be visualized through customized figures. These figures all access the same data basis - the data model.

- Multiple figures can not contradict each other, but they can offer different views. They can focus different details or hide some elements.



This is no contradiction to the example before.
A Cabrio is also a car and may have a parking assistent.

Data basis („the truth“)

- Point
  - Point
  - AlertCPoint
    - alertCLocationCountryCode
    - alertCLocationTableNumber
    - alertCLocationTableVersion
  - Point
  - PointByCoordinates
    - bearing

Elements in one package

Connections / relations

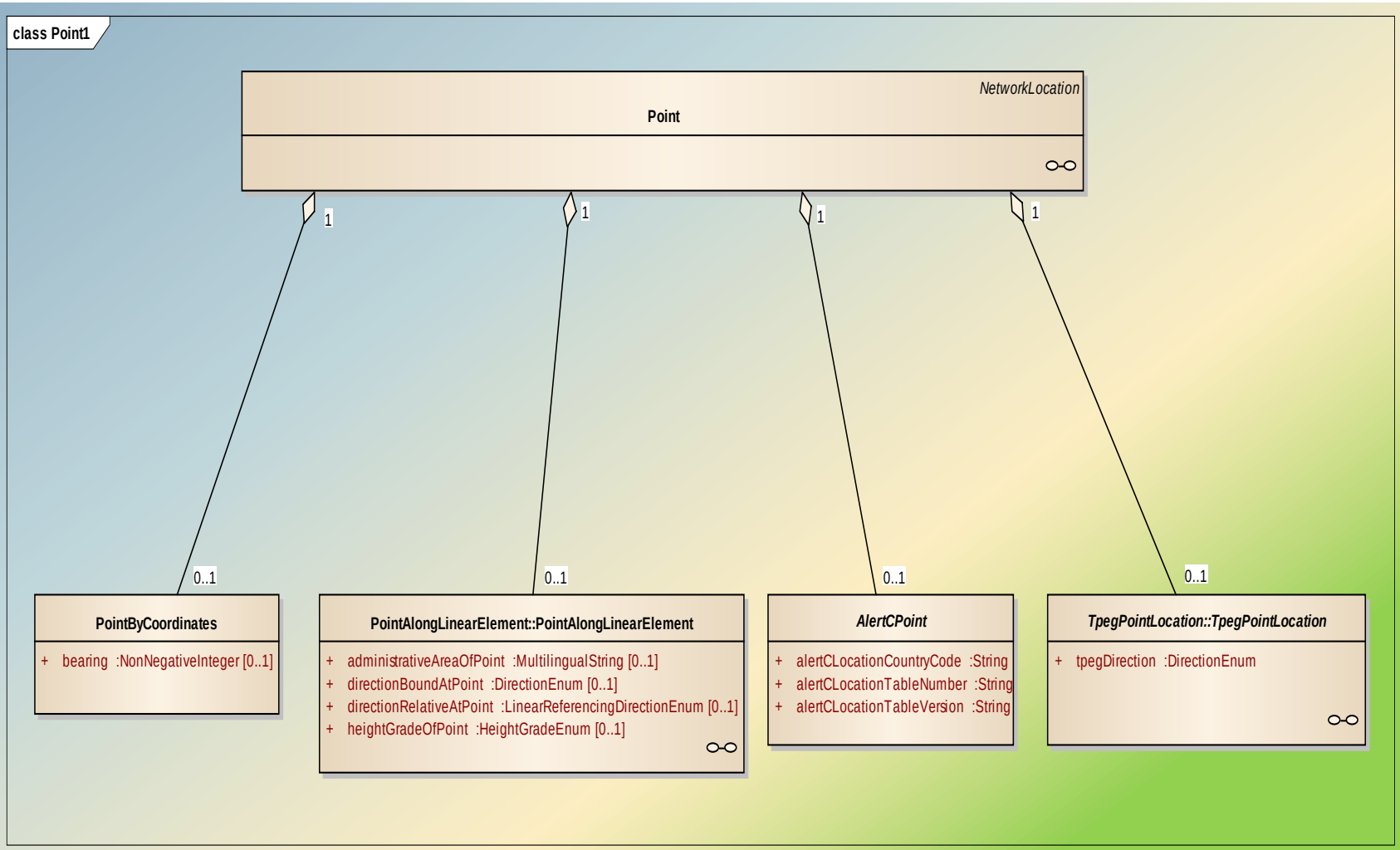
Class : Point

- Properties
  - General
  - Details
  - Templates
  - Advanced
  - Tagged Values
- Rules
  - Requirements
  - Constraints
  - Scenarios
- Related
  - Files
  - Links

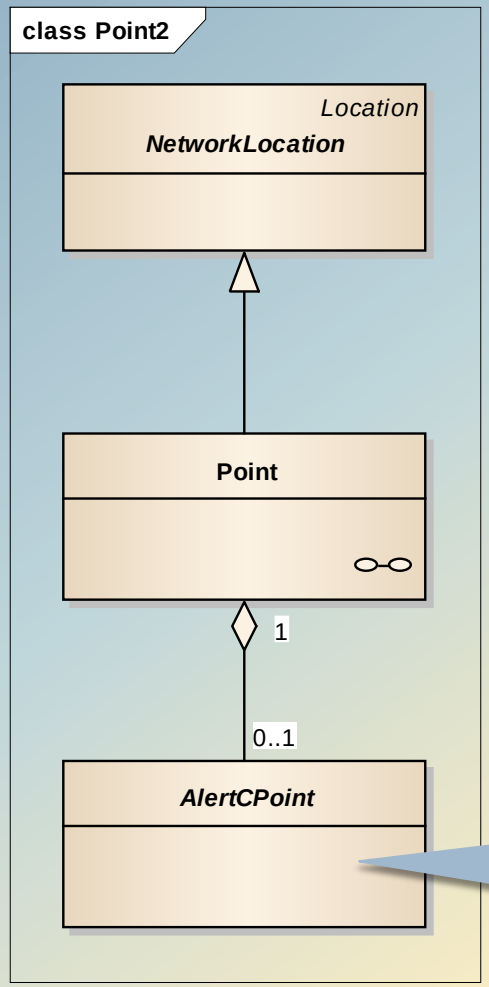
Relationships

Element	Element Stereotype	Type	Connection
AlertCPoint		Class	Aggregation
AssignedParkingSpaces		Class	Aggregation
AssignedParkingSpaces		Class	Aggregation
NetworkLocation		Class	Generaliza...
ParkingFacility	versionedIdentifiable	Class	Aggregation
ParkingFacility	versionedIdentifiable	Class	Aggregation
PointAlongLinearElement		Class	Aggregation
PointByCoordinates		Class	Aggregation
PointDestination		Class	Aggregation
TpegPointLocation		Class	Aggregation

Views on the Point item



Views on the Point item

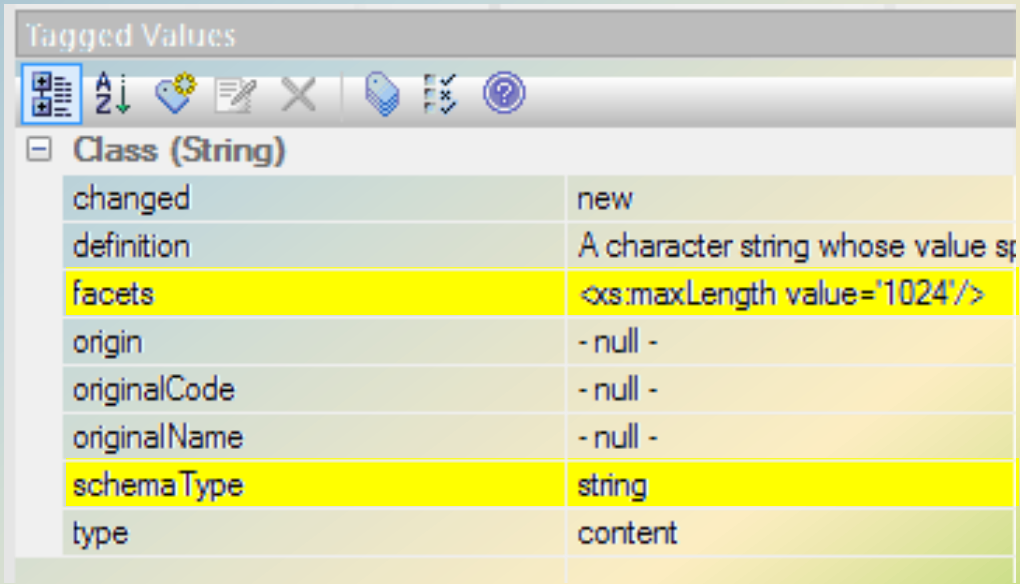


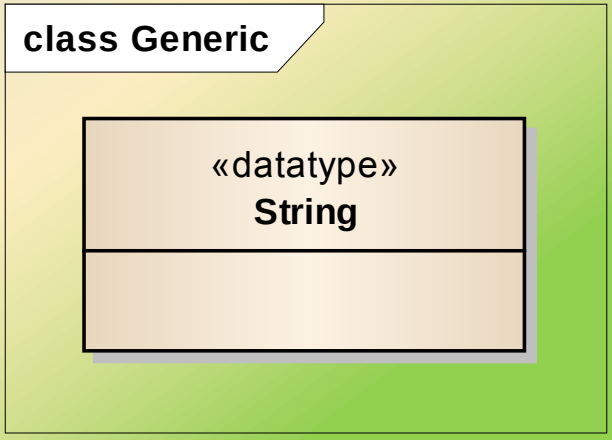
In this view, the possibility to express a point by coordinates is not shown, however, this is still a valid method.

A little awkward, because mandatory attributes are not shown. Anyway, it's not forbidden.

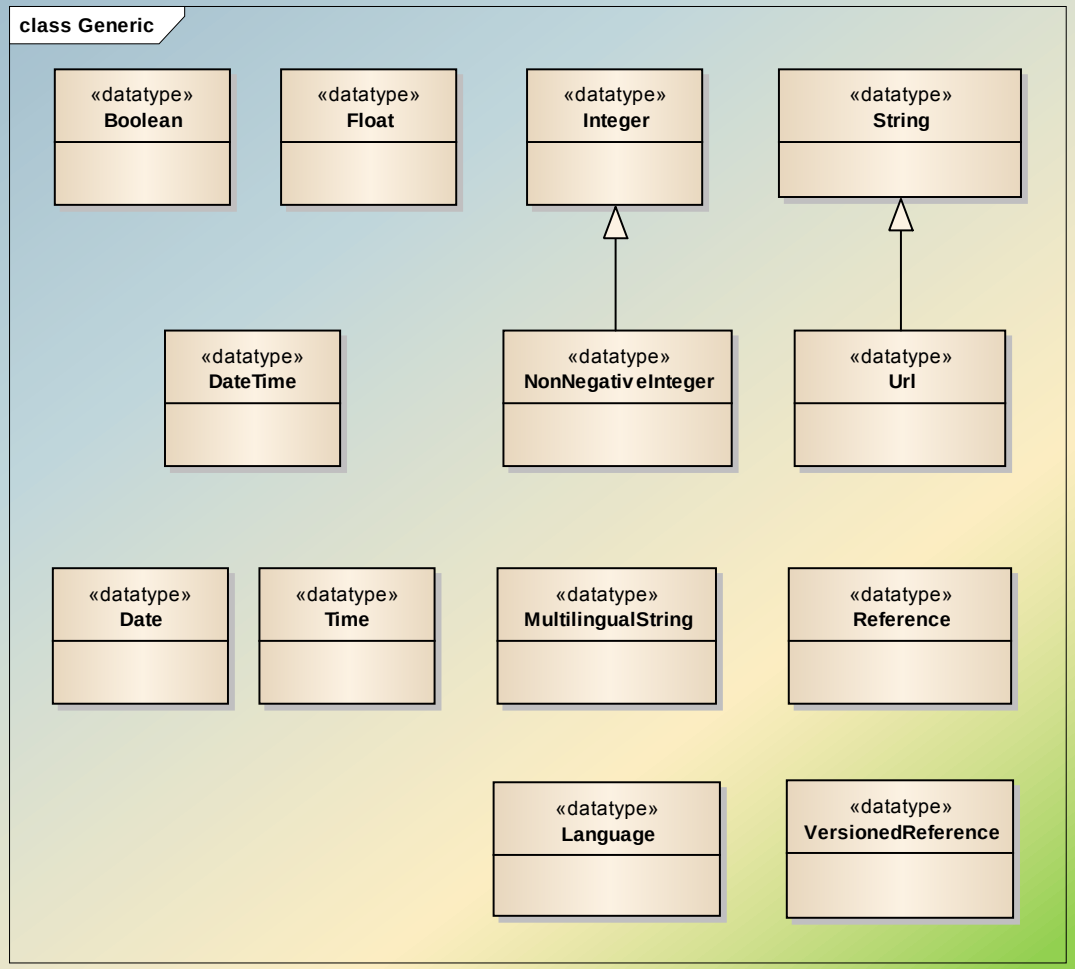
Data types

All used (simple) data types are also realized in the form of DATEX components.
A distinction is made between generic and specific data types. The latter are derived from the generic types.
DATEX simple data types are mapped to the corresponding XSD simple types (using the tagged value 'schemaType').

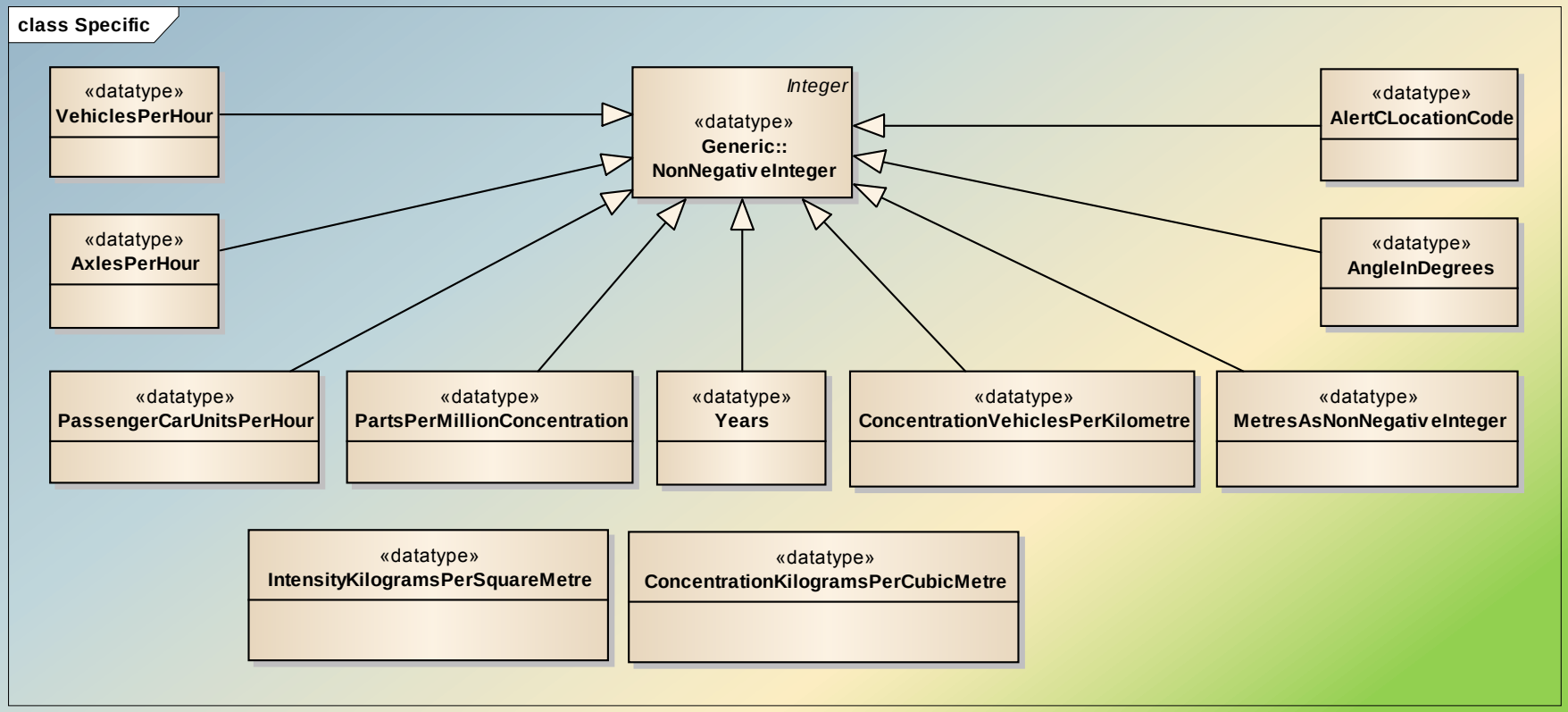




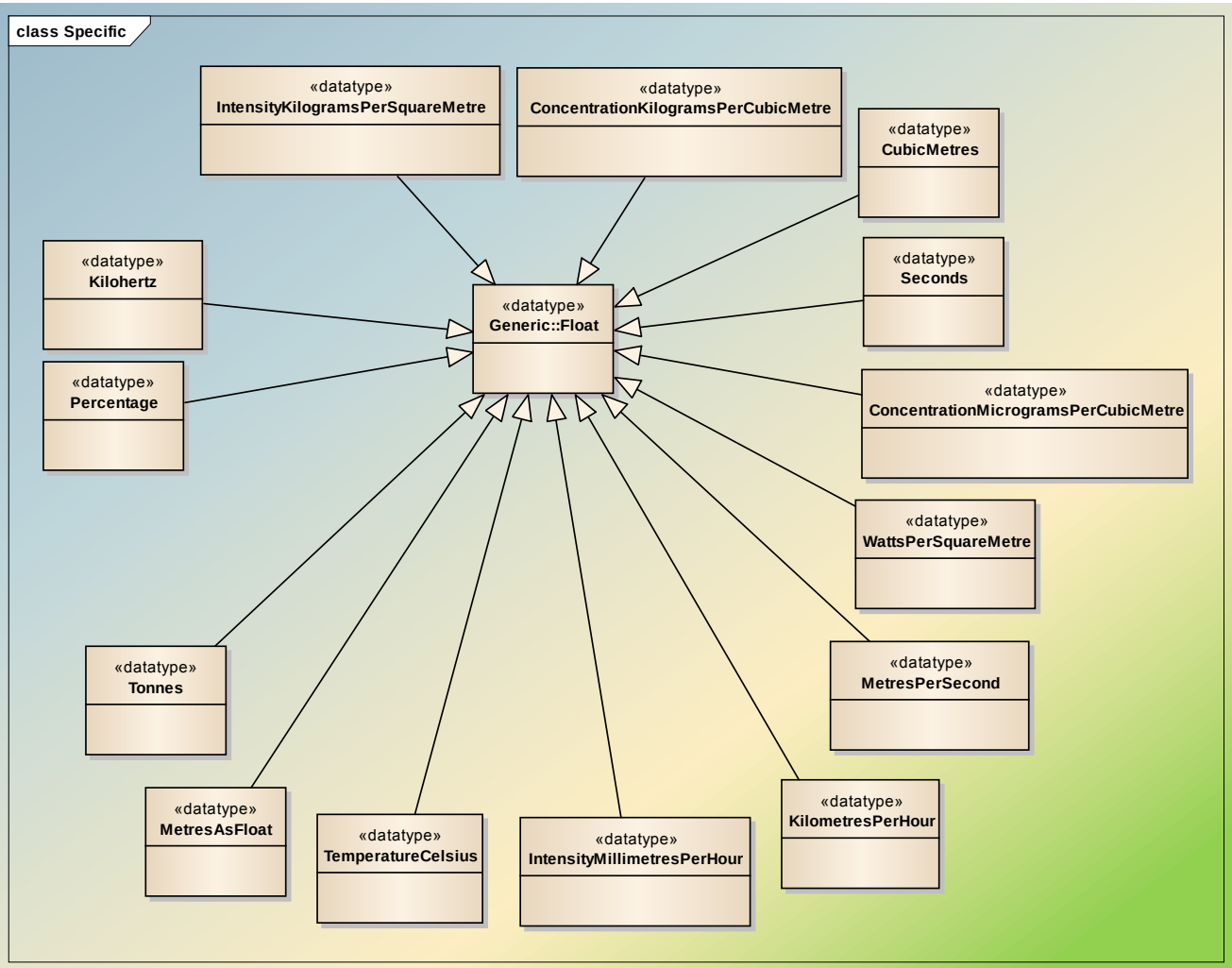
Data types (generic)



Data types (specific I)

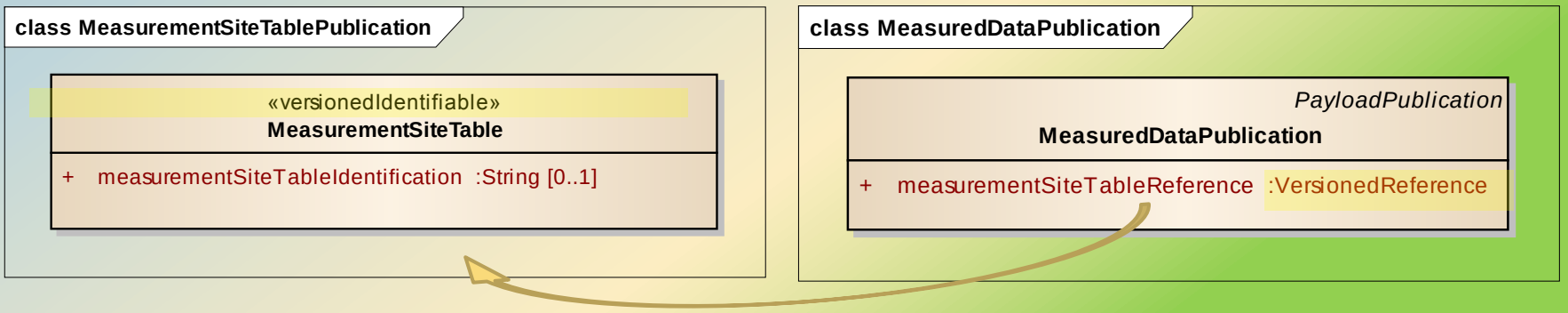


Data types (specific II)



References

- Using cross references as an alternative to direct aggregation or between different messages
- Example: Relationship between static data (e.g. infrastructure information) and dynamic data (e.g. high-frequency measurements)
- Identification by stereotype ‘identifiable‘ (using an ID "unique in space and time")
- Reference with data type ‘Reference ‘
- Different versions: Usage of ‘versionedIdentifiable‘ and 'VersionedReference '.



References in XML instance example

Definition of MeasurementSiteTable in static message:

```
<measurementSiteTable id="92126FC7-3D2E-4AAE-A1AC-FE33812D572F" version="1">
    <measurementSiteTableIdentification>Diebg12</measurementSiteTableIdentification>
    <measurementSiteRecord id="C69BE0FB-CA4C-43CE-A00F-9B3A77E5CB86" version="1">
        <measurementEquipmentTypeUsed>
```

Reference in dynamic message:

```
<payloadPublication xsi:type="MeasuredDataPublication" lang="de">
    <publicationTime>2012-08-14T09:00:01.0Z</publicationTime>
    <publicationCreator>
        <country>de</country>
        <nationalIdentifier>DE-MDM-xxxxxxxx</nationalIdentifier>
    </publicationCreator>
    <measurementSiteTableReference targetClass="MeasurementSiteTable" id="92126FC7-3D2E-4AAE-A1AC-FE33812D572F" version="1"/>
    <headerInformation>
        <confidentiality>noRestriction</confidentiality>
        <informationStatus>test</informationStatus>
    </headerInformation>
```

a so called UUID was used (Universally Unique Identifier)

CREATING PROFILES

What is a DATEX II profile?

- The DATEX data model (Level A) is complex and very extensive.
- There is a large complete scheme (currently more than 15000 rows).
- Communication systems usually require only a small part of the data model:
  - Certain type of message
  - Not all optional attributes
  - Only certain enumerated values

Software costs often correspond to the number of classes
→ aim to minimize

- Therefore it is useful to implement only the actually required elements in the software
- → Tailored XML schema
- With respect to a particular use case this is called a DATEX II profile

What is a DATEX II profile?

- Thus a DATEX II profile is a selected subset of the DATEX Level A model
- But it can also contain additional extensions from Level B
- Due to the backward compatibility of extensions:
  A communication partner who is able to communicate on the entire Level A, can always interpret any valid DATEX profile, even if there is a Level B extension included (in fact, this part is ignored then, but it does not fail)

Examples for DATEX II profiles

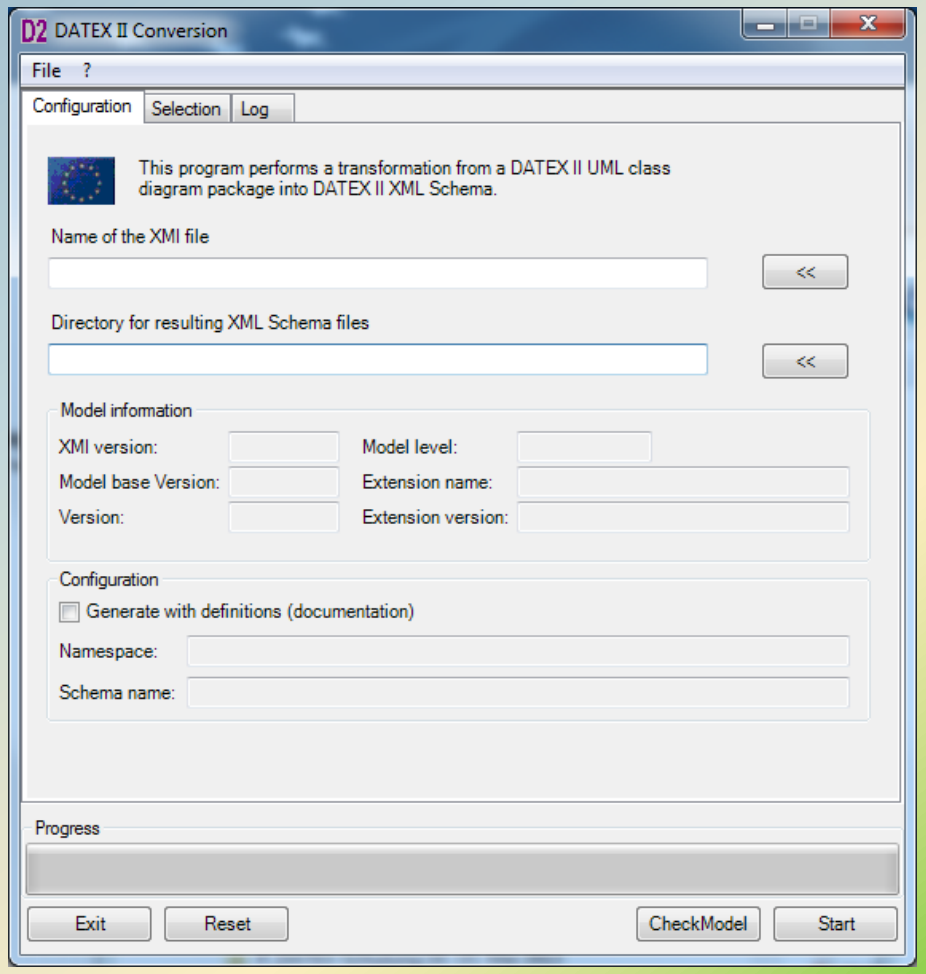
- Profile for Truck Parking
- Profile for Traffic Messages
  (for the German Mobility Data Market Place MDM)
- Profile for Strategic Routing
  (which is a Level B extension)

A DATEX II profile is made up of a tailored XML-schema file and at best a documentation.

The DATEX Tool („DATEX II Conversion“) …

- was developed and is maintained by the DATEX Community
- is working on an XMI representation of the data model (export function in EA)
- verifies compliance with all rules of the meta model (see next slides)
- enables the selection of the desired data
- generates the customized schema file

The DATEX Tool



Selected rules of methodology, here the example for attributes:

Methodology

5.2.3. UML Classes in DATEX II models may have UML Attributes.

5.2.4. UML Attributes shall have a “definition” UML TaggedValue.

5.2.5. UML Attributes shall have an assigned “type” element. The assigned type shall be a UML Class with UML Stereotype “datatype” (Note that built-in UML types are not allowed.) or it shall be a UML Enumeration. If the assigned type is either “Reference” or “VersionedReference”, the UML Attribute shall have a “targetClass” UML TaggedValue, which shall provide a name of a UML Class that has an “identifiable” or “versionedIdentifiable” Stereotype assigned, respectively.

5.2.6. UML Attributes shall have an “order” UML TaggedValue. This order shall be a non negative integer and all order values of attributes of the same UML Class shall be unique within this UML Class.

5.2.7. UML Attributes may have a “multiplicity” element attached. In case multiplicity is not provided explicitly, a default value of "1..1" is used.

5.2.8. UML Attributes names have a global name scope in DATEX II, i.e. two UML Attributes with the same name shall have the same definition and type values.

Selected rules of methodology, here the example for attributes:

Methodology

5.2.3. UML Classes in DATEX II models may have UML Attributes.

5.2.4. U

5.2.5. U

"VersionedIdentifiable" Stereotype assigned, respectively.

5.2.6. U

5.2.7. UML Attributes may have a "multiplicity" element attached. In case multiplicity is not

5.2.8. U

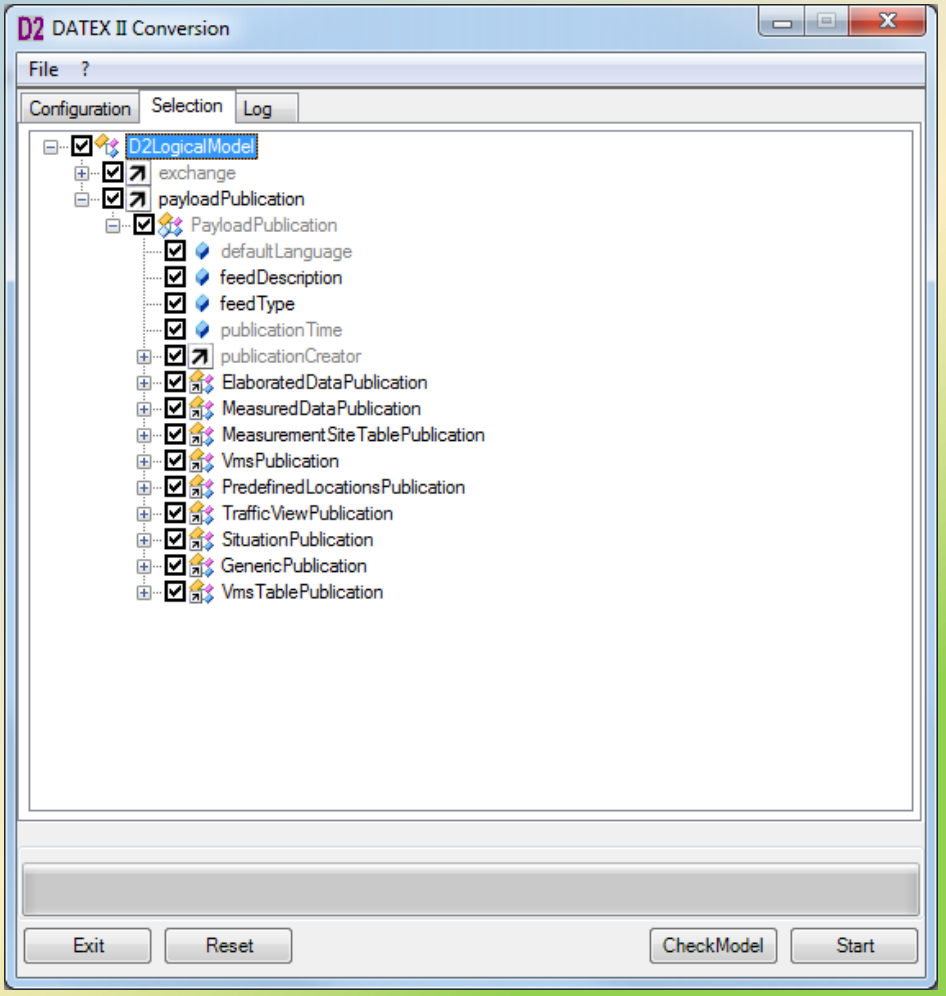
Classes may have attributes. Attributes need a tagged value ‚definition' (as well as all other DATEX elements, too)

Attributes need a tagged value ‚order' as well as a multiplicity

If an attribute appears more than once, name, definition and type must be identical

Three possibilities of selection

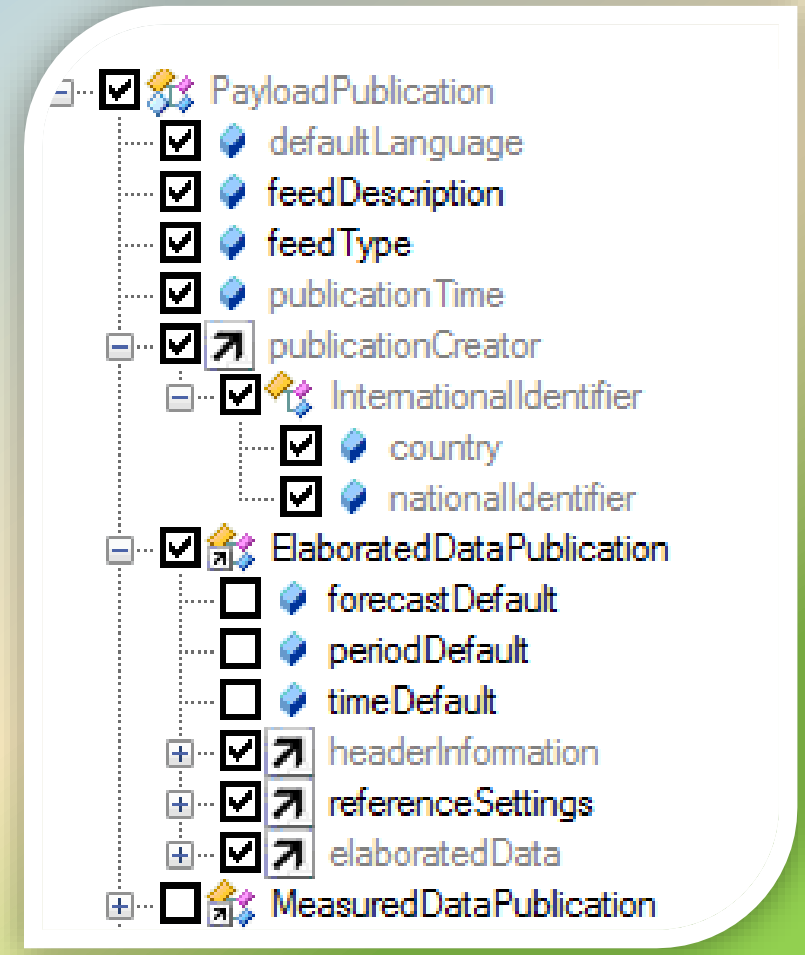
1. Selection of components
2. Modification of multiplicities
3. Selection of enumeration literals



1) Selection of components

Naturally, this is the strongest possibility of reduction because all types of messages and / or message parts may be omitted

- Mandatory elements or parts (grey) can not be deselected
- Parts of multiple use can only be either completely dropped or allowed anywhere (improvement at this point is planned)

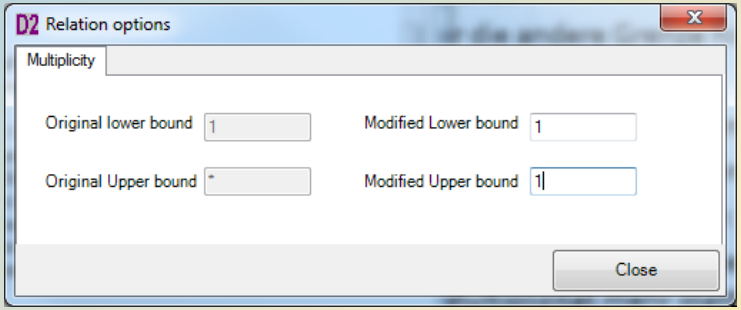


2) Modification of multiplicity of attributes or aggregations

- Possible since DATEX II version 2.1
- To keep compatibility to Level A, only the following changes are possible:
  - Increasing the lower boundary
  - Decreasing the upper boundary
  - … but in each case not about the other limit.

Examples:

- An optional attribute can be set mandatory: 0..1 → 1..1
- An unlimited attribute can be limited: 0..n → 0..3
- Limitation can be made stronger: 1..4 → 2..3
- But for a simple mandatory attribute (1..1), multiplicity cannot be modified any more.



Visualisation of this issue in Enterprise Architect is difficult, because the Level A model would be changed.

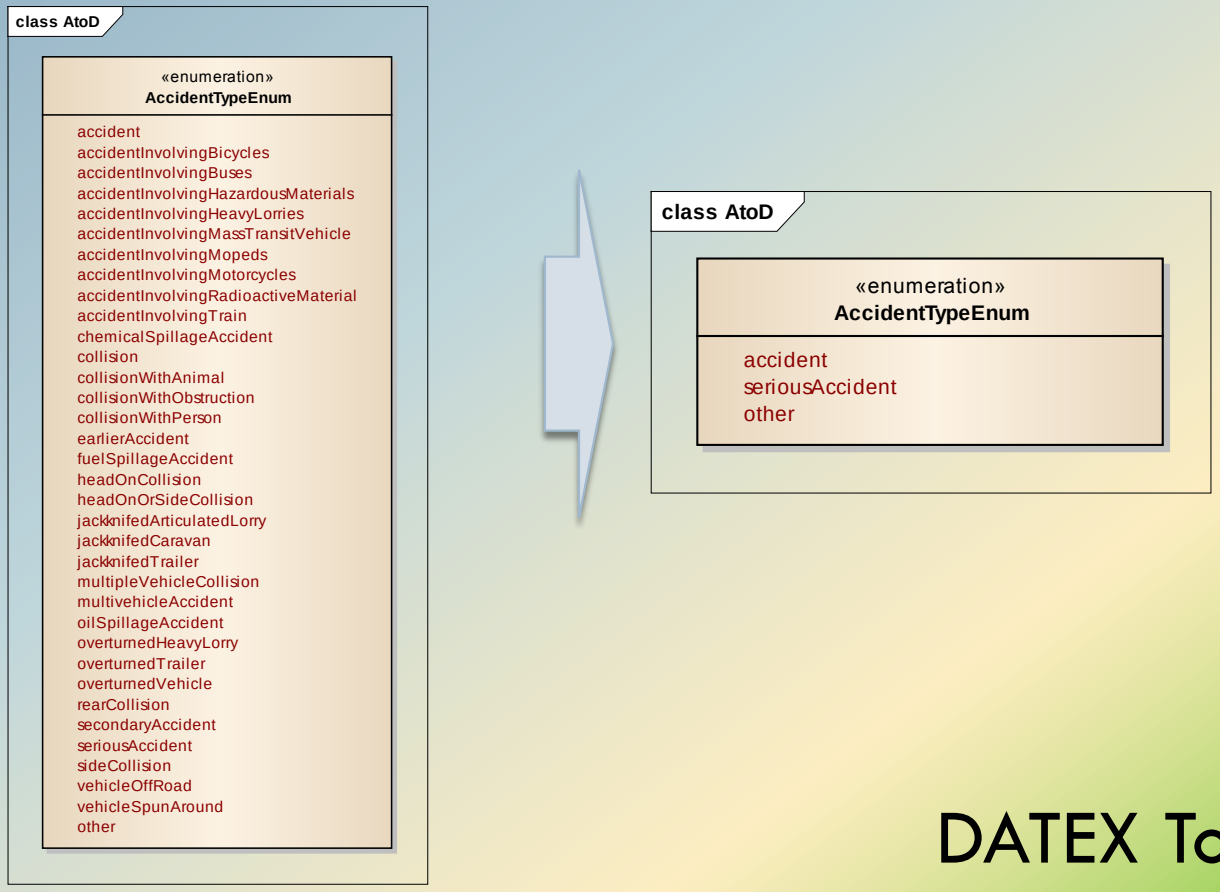
3) Selection of enumeration literals

- Possible since DATEX II version 2.1
- The DATEX II data model contains a lot of enumerations, some of them with a huge amount of literals.
- Communication systems often only use few literals, e.g. only ‘accident‘ and 'serious accident‘
- Keeping the literal ‚other‘ is useful for later changes or extensions.

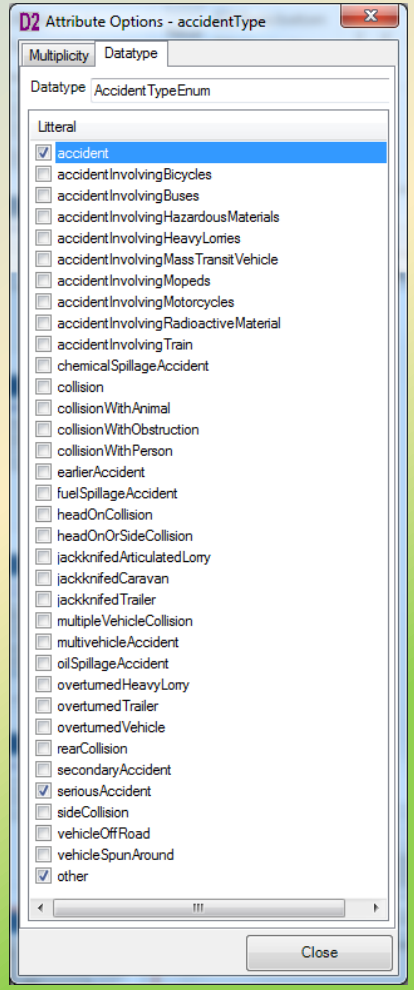
Visualisation of this method is possible in Enterprise Architect because in every figure attributes and literals can be made (in)visible.

3) Selection of enumeration literals

Enterprise Architect



DATEX Tool



Profiling

- The applied selection (in all three mentioned forms) can be saved with the tool and can be reloaded (i.e. applied) at a later point of time.
- This way a simple reconstruction of the profile is possible at any time.
- Final outcome of the tool is the correspondingly reduced XML schema file (.xsd)

Summary

Summary

- Using the freely available DATEX tool, the Level A model can be adapted to the respective application.
- The result is a slim XML schema that is nevertheless understood by all systems that dominate the Level A model.
- By selections and restrictions, the content that is expected in the message can be focused very precisely.
- By limiting the elements actually needed, the software development costs can be kept as low as possible.
- Profiles can also be extended with level B extensions (see following chapter).

Extensions

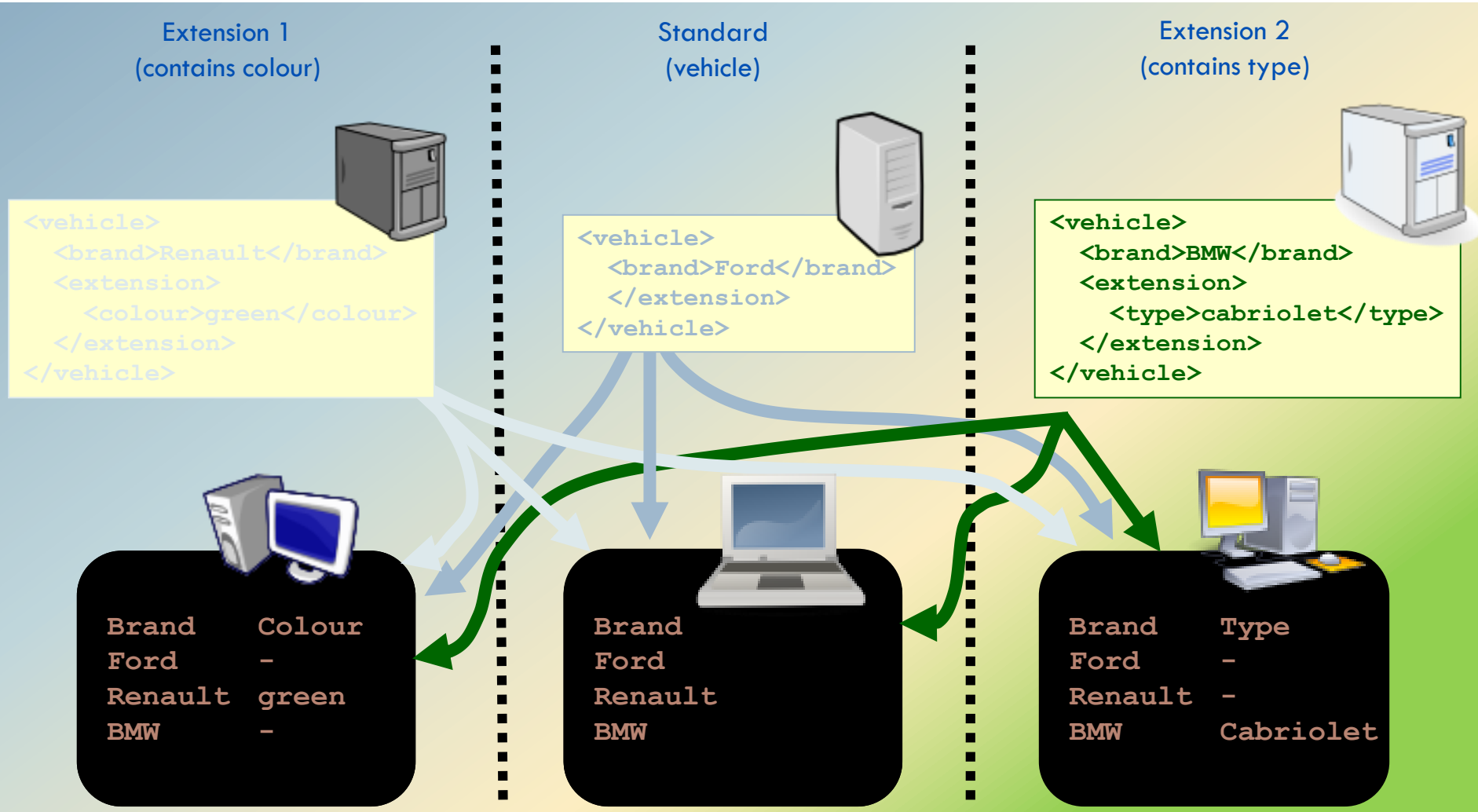
MODELING AND USAGE OF EXTENSIONS

European Commission
Mobility & Transport

DATEX data model ‚interoperability levels‘

- Level A
  - ‚Existing' data model
  - Data catalogue, ontology, data registry, ….
  - The Level A model is 'fix' and can only be adjusted within new DATEX versions, but not in general
    (but see also profile creation)
- Level B
  - Backward compatible extension/amendment to Level A
- Level C
  - Completely independent, no longer to Level A compatible scheme which corresponds to the DATEX methodology anyway.

Level B extensions: The principle



Rules for extensions

Methodology

a) A model that is conforming to this Technical Specification may be ex… Extensions may either seek backwards compatibility to an existing model (de… 'core model' in this clause), or they may create a new model not compatible to any previous model, but nevertheless using the methodology provided within this Technical Specification and – potentially – reusing classes taken from other, existing models. A compatible extension is denoted within this Technical Specification as a level B extension. Non-compatible extensions are denoted as level C extensions.

b) All extensions shall fully comply with all other rules presented so far in this document.

c) An extended model shall provide extension name and version number in two tagged values called "extensionName" and "extensionVersion" on the “d2LogicalModel” element and on any other root level elements (defined using a “rootElement” tagged value), that shall be usable in conjunction with extended elements.

Rules for extensions

Methodology

A model that is conforming to this Technical Specification may be ex
Extensions may either seek backwards compatibility to an existing model (de

Level B is backward compatible.
Level C not.

b) All extensions shall fully comply with all other rules presented so far in this document.

c) Extensions must follow the basic rules for Level A.

element and on any other root level elements (defined using a "rootElement" tagged

The tagged values ‚ExtensionName' and ‚ExtensionVersion' are to be used.

Rules for extensions

Methodology

Classes belonging to an extension and having a superclass not belonging to the extension (i.e. extension classes that inherit from the core model) shall have an "extension" tagged value with values either "levelb" or "levelc". Extensions that do not add new root classes (i.e. classes that have a “rootElement” tagged value) are called “level B extensions”. These extensions shall set the “extension” tagged values to “levelb”. They are backwards compatible with the standard model on message level. Extensions that introduce new root classes are called “level C extensions” and shall set the “extensions” tagged value to “levelc”.

e) Classes belonging to an extension may not become superclasses of classes in the core model, i.e. specializations from a class from the extension to a class in the core model may not be added to the model.

f) UML Associations may be added to the extended model that have a core model class on their source end and an extension class on their target end. Thus, existing classes from the core model may become components from containers in the extensions model (class reuse), but classes from the extensions shall not become components of existing containers in the core model.

g) Data types and enumerations of the core model may be reused in extensions.

Rules for extensions

Methodology

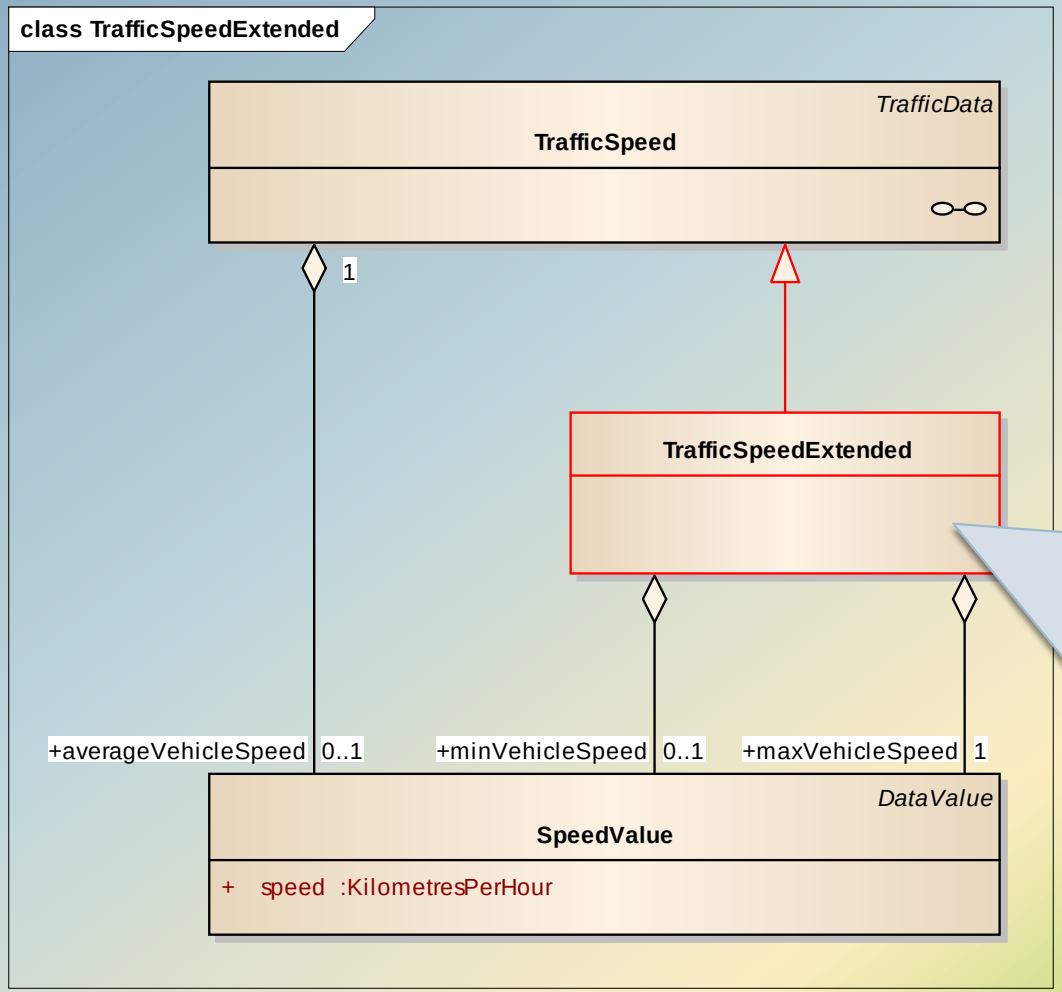
The base class of the extension must get a tagged value ‚extension' – ‚levelb' or ‚levelc'.

Extension classes must not be a superclass for a Level A component.

Extension classes may use existing Level A classes (aggregation), but not vice versa

Enumerations from Level A may be used.

Level B Extension



This is no generalisation/ specialisation, although the same symbol – a sharp arrow – is used.

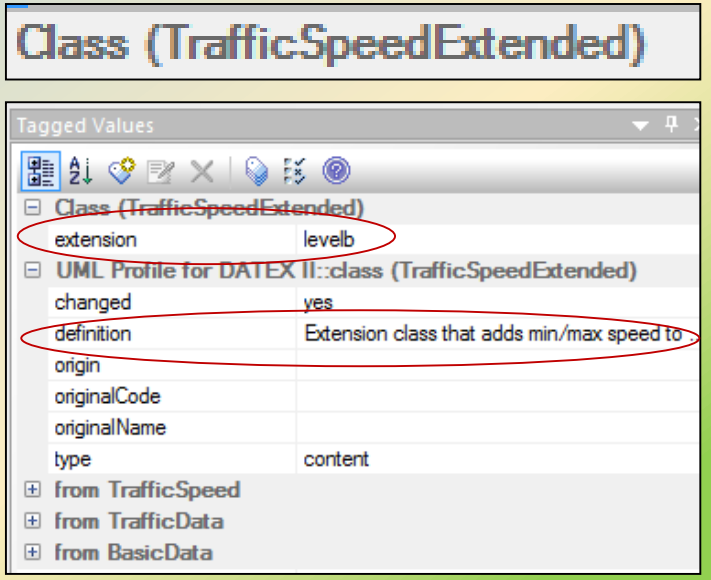
For a better distinction, it is possible to colour it in red, for example (informal).

The DATEX tool will identify the difference by the tagged value “extension” – “levelb”

Mandatory tagged values (also in Level A)

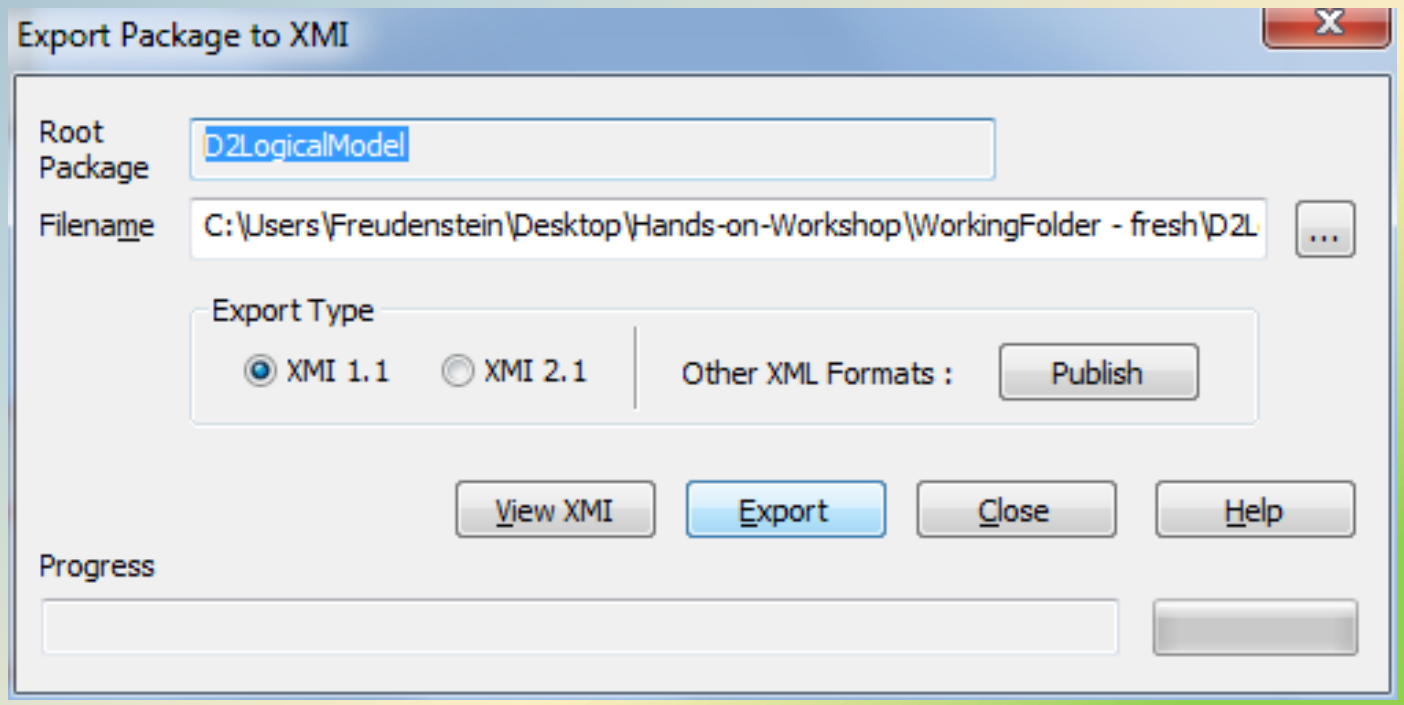
- "definition" - text
  - for every package
  - for every component (including enumerations)
  - for every attribute
  - for every literal
  - for aggregations, if there are two aggregations connecting the same objects (‚parallel')

  "order" – number, must be unique within this object
  - for every attribute
  - for every literal
  - for every aggregation (at the ‚target')
- "extension" – "levelb"
  - Only for the root components of a level B extension (not their successors)
- "extensionName" and "extensionVersion"
  - Only in component „D2LogicalModel", if the model is extended
- „targetClass" – <name of the target component>
  - Only for attributes of type „Reference" or „VersionedReference". The type of the target component has to be specified there.

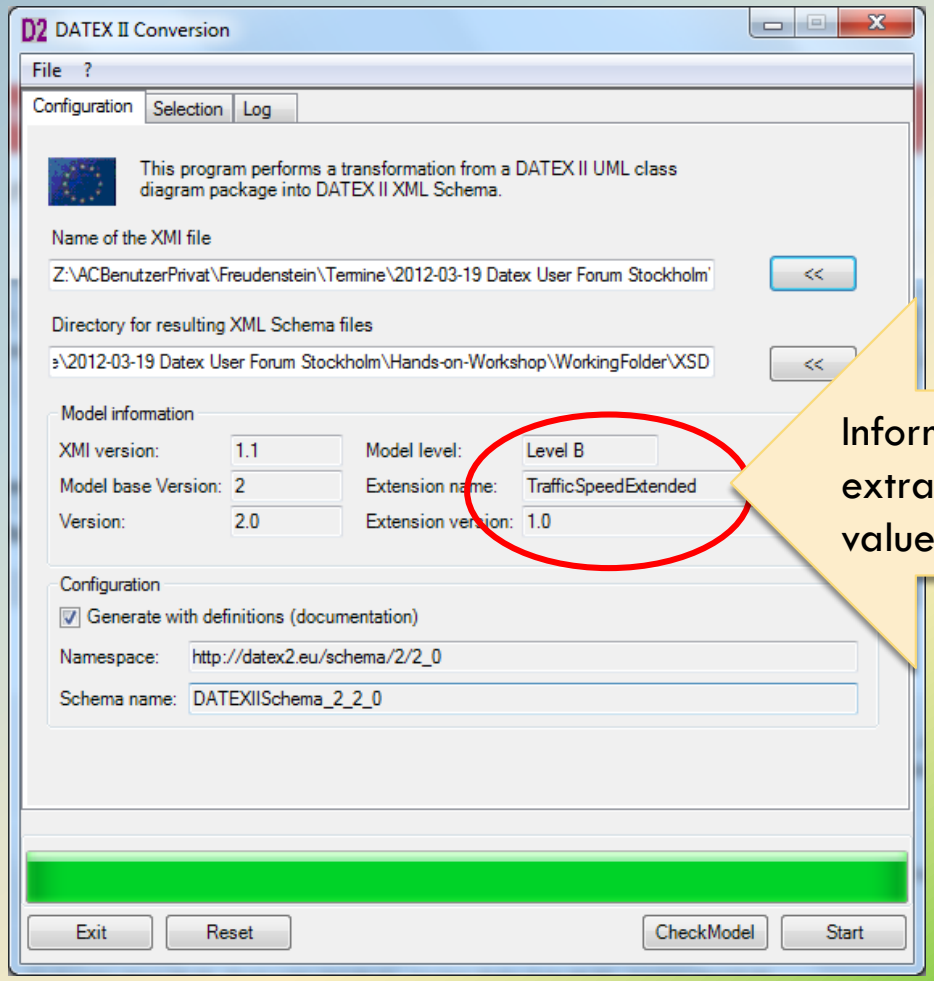


XMI export in EA

- The exchange format between UML and the DATEX Tool is the standardized XMI (1.1)
- The complete package “D2LogicalModel” has to be exported.

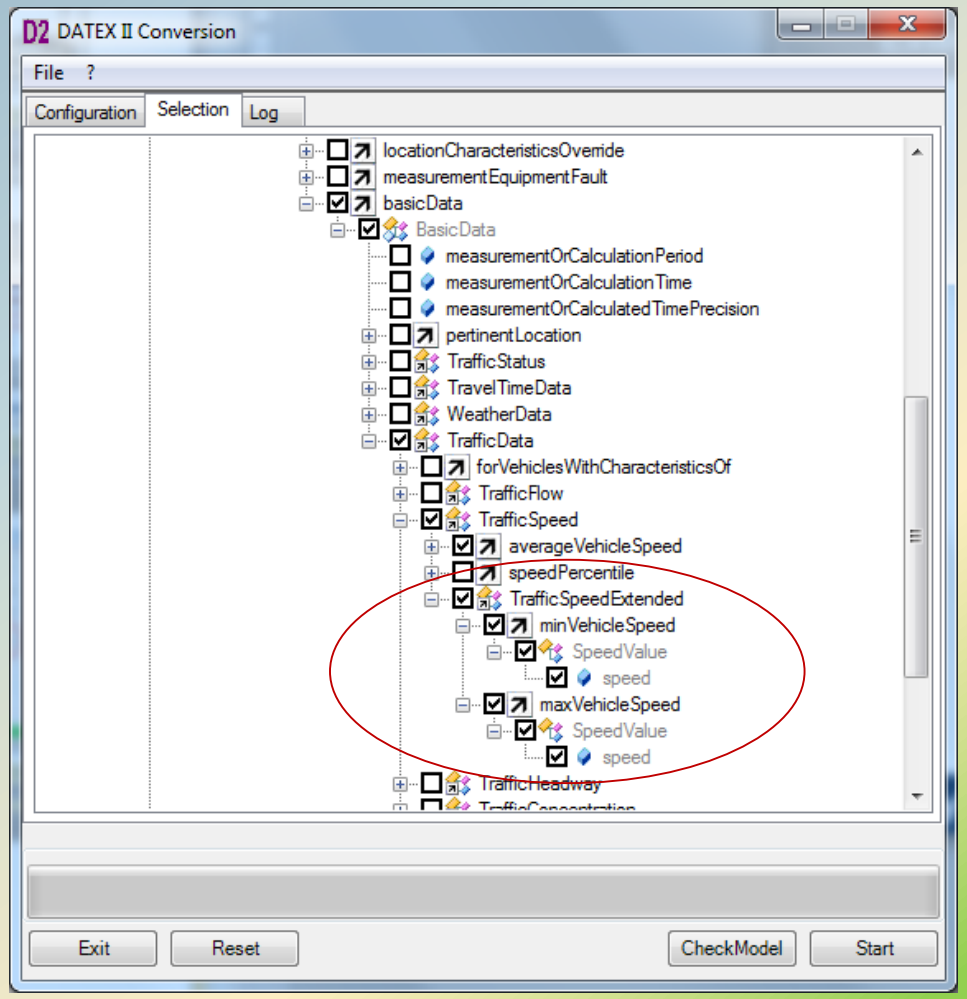


Extension



Information is automatically extracted from the tagged values

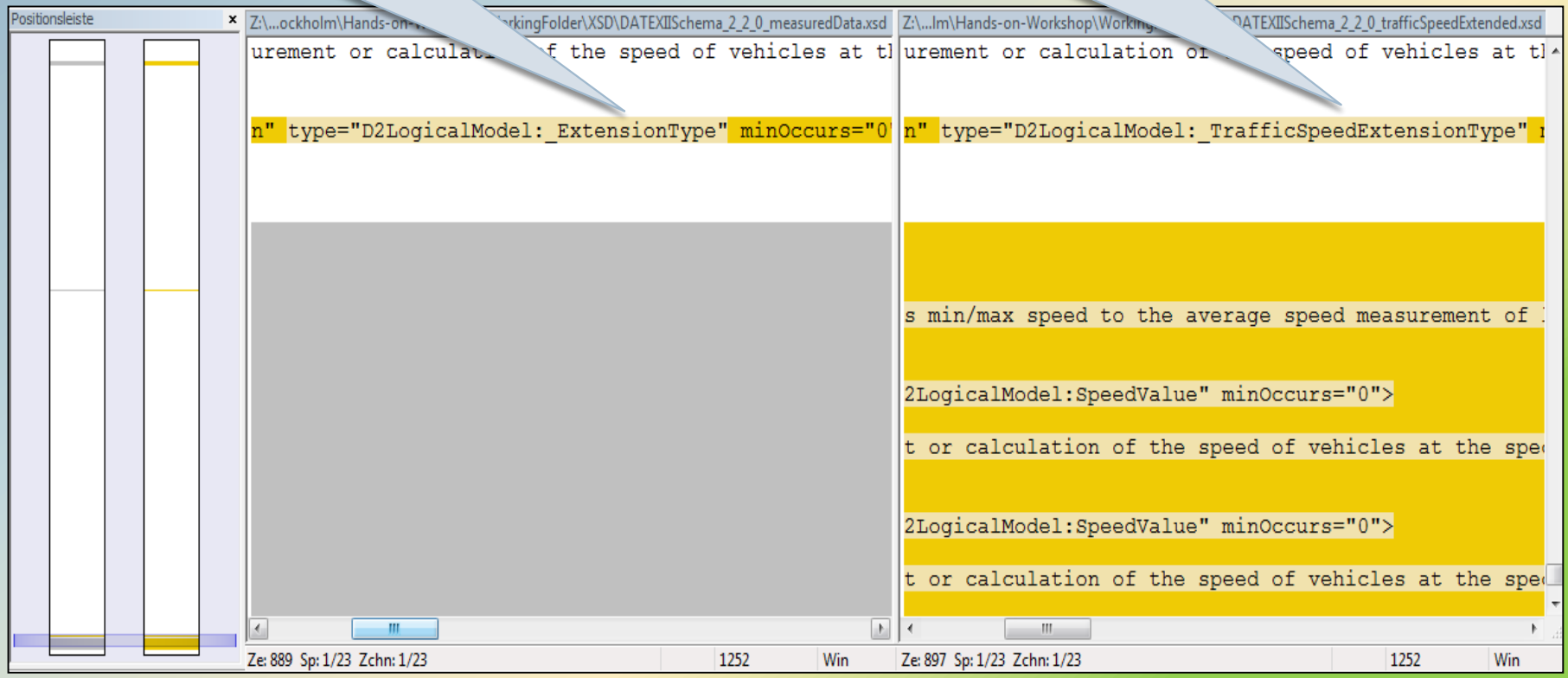
Selection of the extension in the official DATEX Tool



Comparison extended schema (right side) with not extended (left side) schema

Every component points to a default _ExtensionType

In Level B extensions, this is extension is specified individually



XML instance with additional values in the extension part

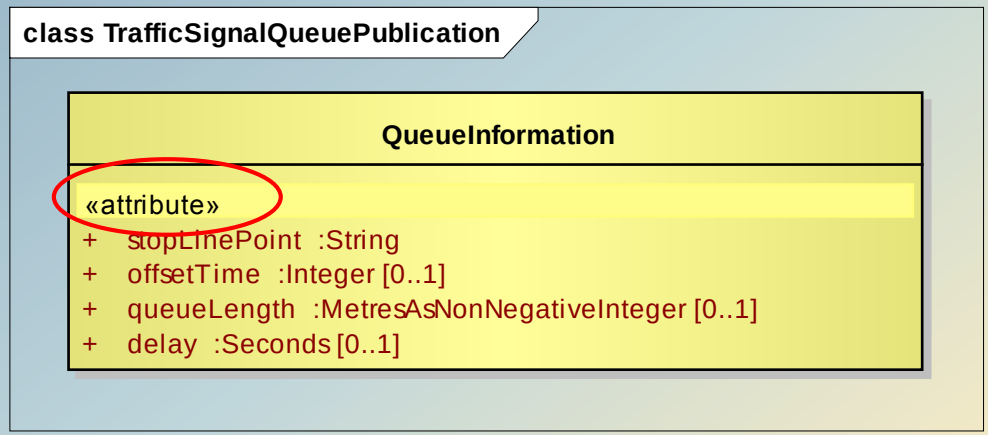

```
<D2LogicalModel:measurementTimeDefault>2013-03-19T09:15:04.0Z</
D2LogicalModel:measurementTimeDefault>
<D2LogicalModel:measuredValue index="0">
  <D2LogicalModel:measuredValue>
    <D2LogicalModel:basicData xsi:type="D2LogicalModel:TrafficSpeed">
      <D2LogicalModel:averageVehicleSpeed>
        <D2LogicalModel:speed>47</D2LogicalModel:speed>
      </D2LogicalModel:averageVehicleSpeed>
      <D2LogicalModel:trafficSpeedExtension>
        <D2LogicalModel:trafficSpeedExtended>
          <D2LogicalModel:minVehicleSpeed>
            <D2LogicalModel:speed>23</D2LogicalModel:speed>
          </D2LogicalModel:minVehicleSpeed>
          <D2LogicalModel:maxVehicleSpeed>
            <D2LogicalModel:speed>61</D2LogicalModel:speed>
          </D2LogicalModel:maxVehicleSpeed>
        </D2LogicalModel:trafficSpeedExtended>
      </D2LogicalModel:trafficSpeedExtension>
    </D2LogicalModel:basicData>
  </D2LogicalModel:measuredValue>
</D2LogicalModel:measuredValue>
```

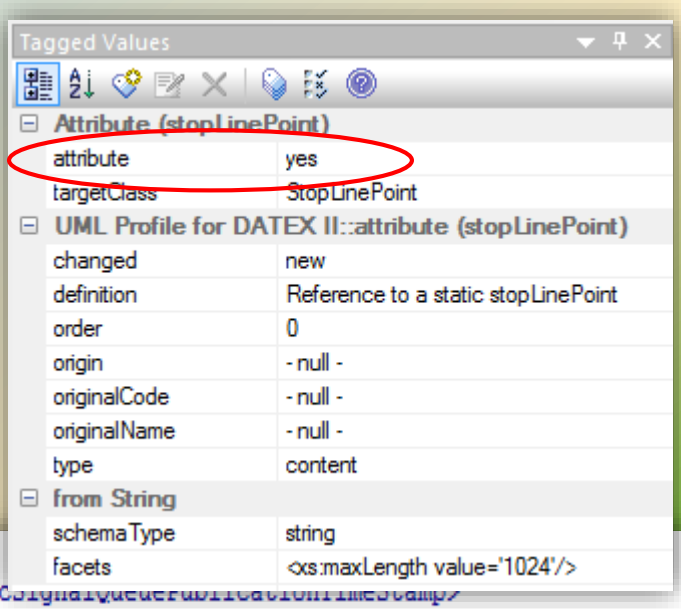
Validation XML against schema

XML instance	XSD (schema)	validates?
Without bug in XML	TrafficSpeedExtended	Yes
	D2	Yes
With bug in Level A	TrafficSpeedExtended	No
	D2	No
With bug in Level B	**TrafficSpeedExtended**	**No**
	D2	**Yes**

Inline attributes

To obtain particularly slim XML instances (for example, for high-frequency signals) the modelling of inline attributes is possible:






```
<trafficSignalQueuePublication>
    <trafficSignalQueuePublicationTimeStamp>2012-10-05T09:01:12.0Z</trafficSignalQueuePublicationTimeStamp>
    <queueInformationValidityTime>2012-10-05T12:00:00.0Z</queueInformationValidityTime>
    <staticTrafficSignalPublicationReference targetClass="StaticTrafficSignalPublication" id="064564C5-4429-4EF8-BF06-B962D6
    <queueInformation stopLinePoint="V501-87C" delay="100" offsetTime="15" queueLength="125" />
    <queueInformation stopLinePoint="V500-84C" delay="90" queueLength="100" />
</trafficSignalQueuePublication>
```

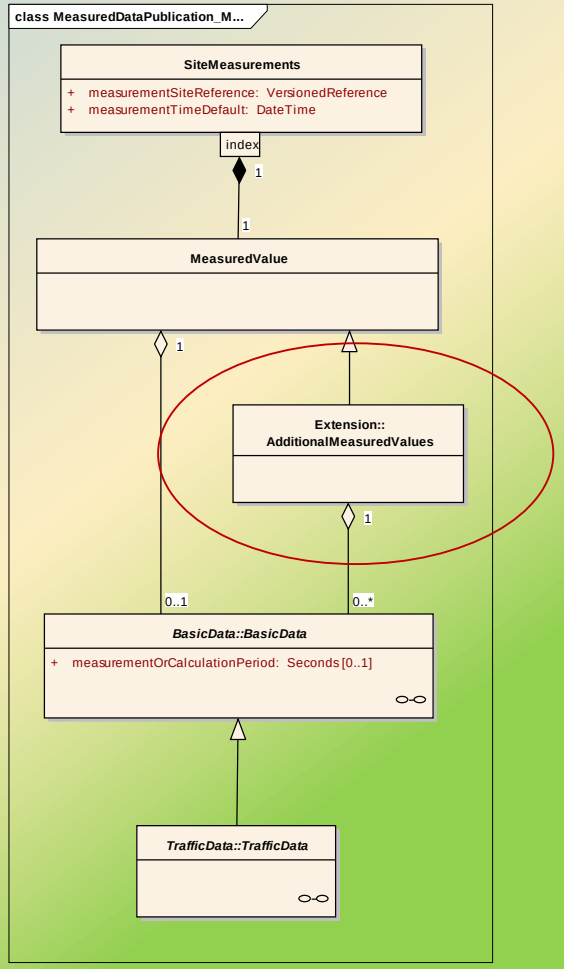
Example of an extension (measurements)

Messtellen Zeitachse →

A	A0	A1	A2	A3	A4	A5				
B	B0		B1		B2		B3		B4	
C	C0			C1			C2			C3
D	D0				D1				D2	
E	E0					E1				
	↓			↓			↓			↓
	A0 B0 C0 D0 E0		A1,A2,A3 B1 C1			A4,A5 B2,B3 C2 D1 E1			B4 C3 D2	

Datengeber: Push- oder Pull-Übertragung mit fixem Übertragungsintervall

MDM-Broker



Published extensions and profiles

- The publication of extensions and profiles on the DATEX website is not mandatory
- But this increases the publicity, if standardisation is desired.

Version: 2.2

Title
Traffic Light Information

Version: 2.1

Title
Italian Motorways Operators DATEX II Profile and Extensions
Travel Times and Traffic Condition
DATEX II profile for Mobile Lane Closure Trailers

Version: 2.0

Title
Data model for TMPs and Navigation systems

Version: 2.2

Title	Country	Organization
Parking Publication Extension v0.5	Germany	AlbrechtConsult GmbH
Linear By Coordinates	Germany	AlbrechtConsult GmbH
OpenLR Extension 1.5	Sweden, Netherlands	Swedish Transport Administration, TomTom International B.V. Dutch National DataWarehouse for Traffic Information (NDW(

Version: 2.1

Title	Country	Organization
VMS Operational Exchange and TMP Management	Italy	IT Motorways DATEX group

Version: 2.0

Title	Country	Organization
LineString extension	Sweden	Viati
Road Infrastructure Publication Extension	Portugal	Armis - Sistemas de Informação, Ida

Version: 2.0 RC2

Title	Country	Organization
Data Quality Publication Extension	UK / England	Highways Agency
CCTV Publications Extension Updated	UK / England	Highways Agency

Version: 2.0 RC1

Title	Country	Organization
VMS Publication Extension	Italy / England	Autostrada per l'Italia and The Highways Agency (England)
Catalogue Extension	Germany	GEVAS software GmbH

1 2 next › last »

Thank you

AlbrechtConsult GmbH

Theaterstr. 24

D 52062 Aachen

+49 241 500 720

Jörg Freudenstein – j.freudenstein@albrechtConsult.com