XML Model or Entity Relationship Data Model:

Which Is Best?

DAMA Phoenix

September 12, 2013

Norman Daoust

Daoust Associates

Introduction

- ➢Daoust Associates, 2001
- ➢consultant, author, speaker
- ➢data modeling and database design
- ➢healthcare electronic data exchange using HL7 standard
- ➢requirements modeling
- ➢business analysis training

Introduction: Presentation Goals

- similarities and differences between entity relationship data modeling and XML modeling

- advantages and disadvantages of both entity relationship data models and XML models

- appropriate usages for data models and XML models

- criteria for determining which to create

Introduction: Presentation Outline

- Why should you care?
- Definitions
- Examples
- Comparison: XML model vs.. entity relationship model
- Deriving XML models from an entity relationship model
- References
- Summary

Introduction: Disclaimers

- informal vs.. dictionary definitions
- illustrative rather than proscriptive
- examples are deliberately incomplete and beyond reproach!

Why should you care?

- XML models are here to stay
- XML models are proliferating rapidly, frequently single-purposed and created in isolation
- XML models are frequently not created by data management professionals
- data management professionals need to be at the forefront

Definitions (1 of 2)

- ER model: “a data model for describing a database in an abstract way” per Wikipedia; for describing data relevant for an organization
  - includes diagram(s) and associated text

- instance: a specific occurrence of an ER model or XML model

- service (as in SOA): a unit of functionality packaged for convenient and consistent use
  - includes set of operations, each with an input and output message

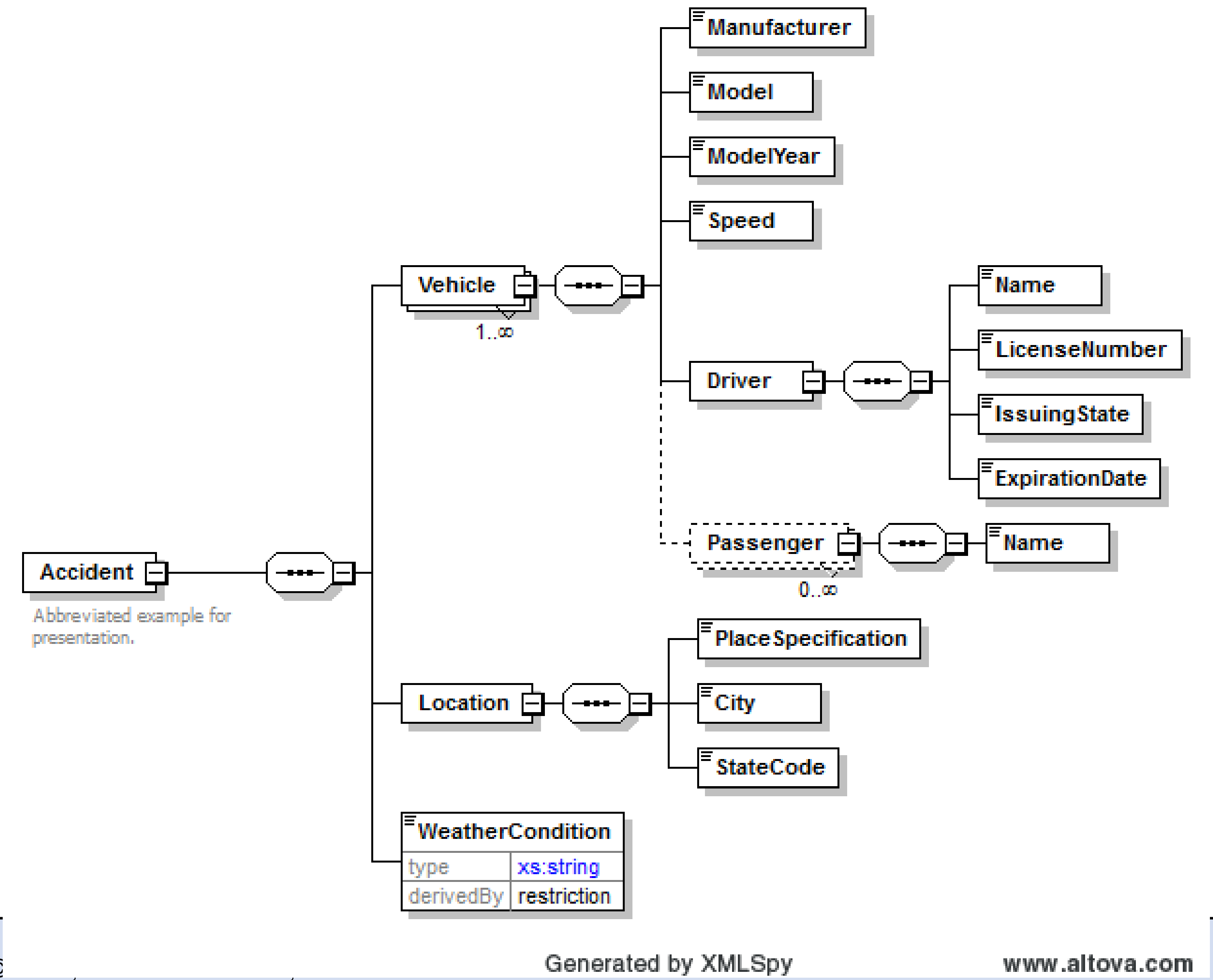
Definitions (2 of 2)

- XML model: a specification for a set of XML documents
  - includes diagram(s) and associated text
- XML model formats: diagram/logical, grid/physical, text
- XML document: an instance of an XML model

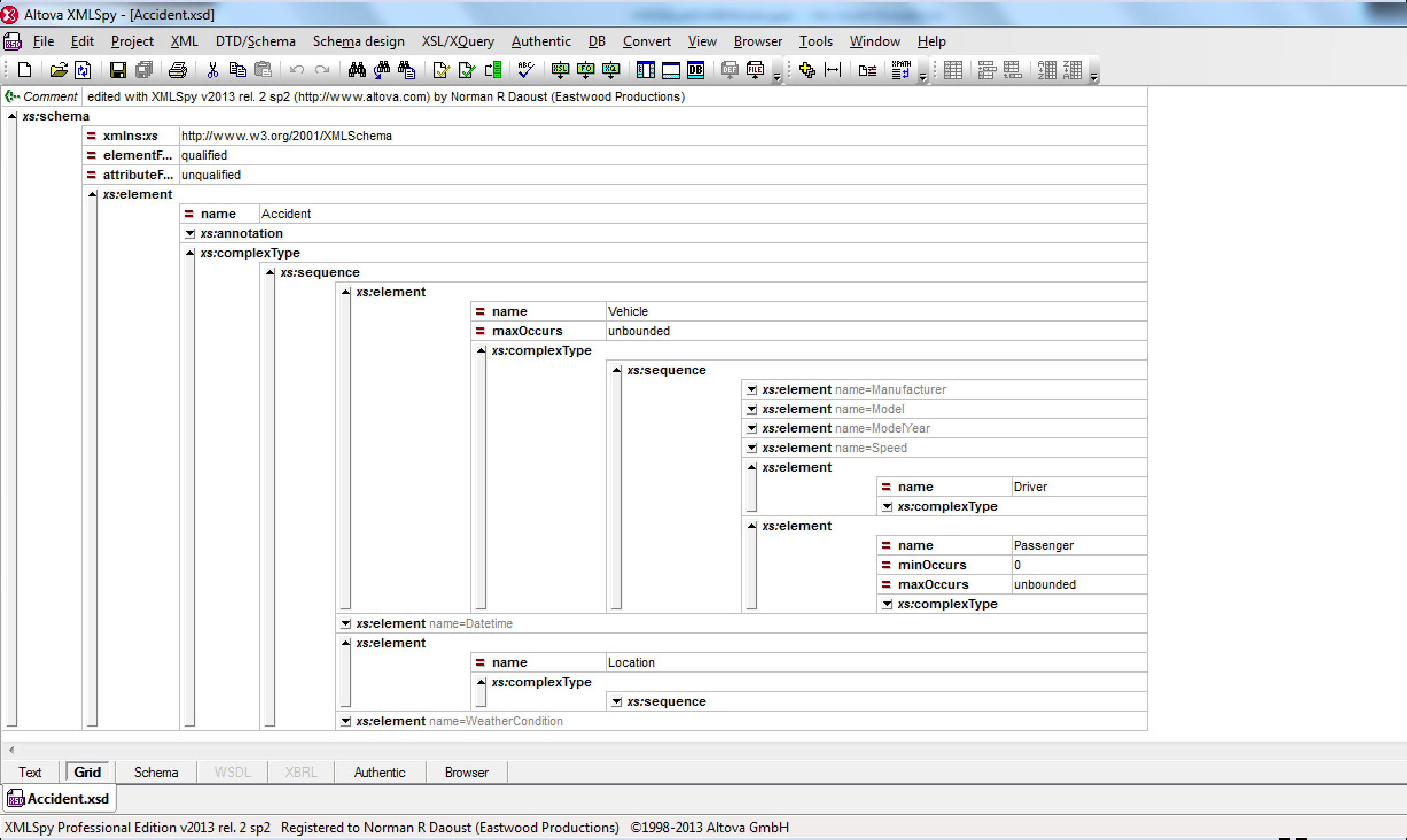
Example

- Accident: motor vehicle
- XML model: diagram, grid, text view
- XML document: text view
- ER model (derived from XML model): diagram, text view
- ER model instance: diagram
- ER model (not derived from XML model): diagram

XML model: diagram representation



XML model: grid/physical representation

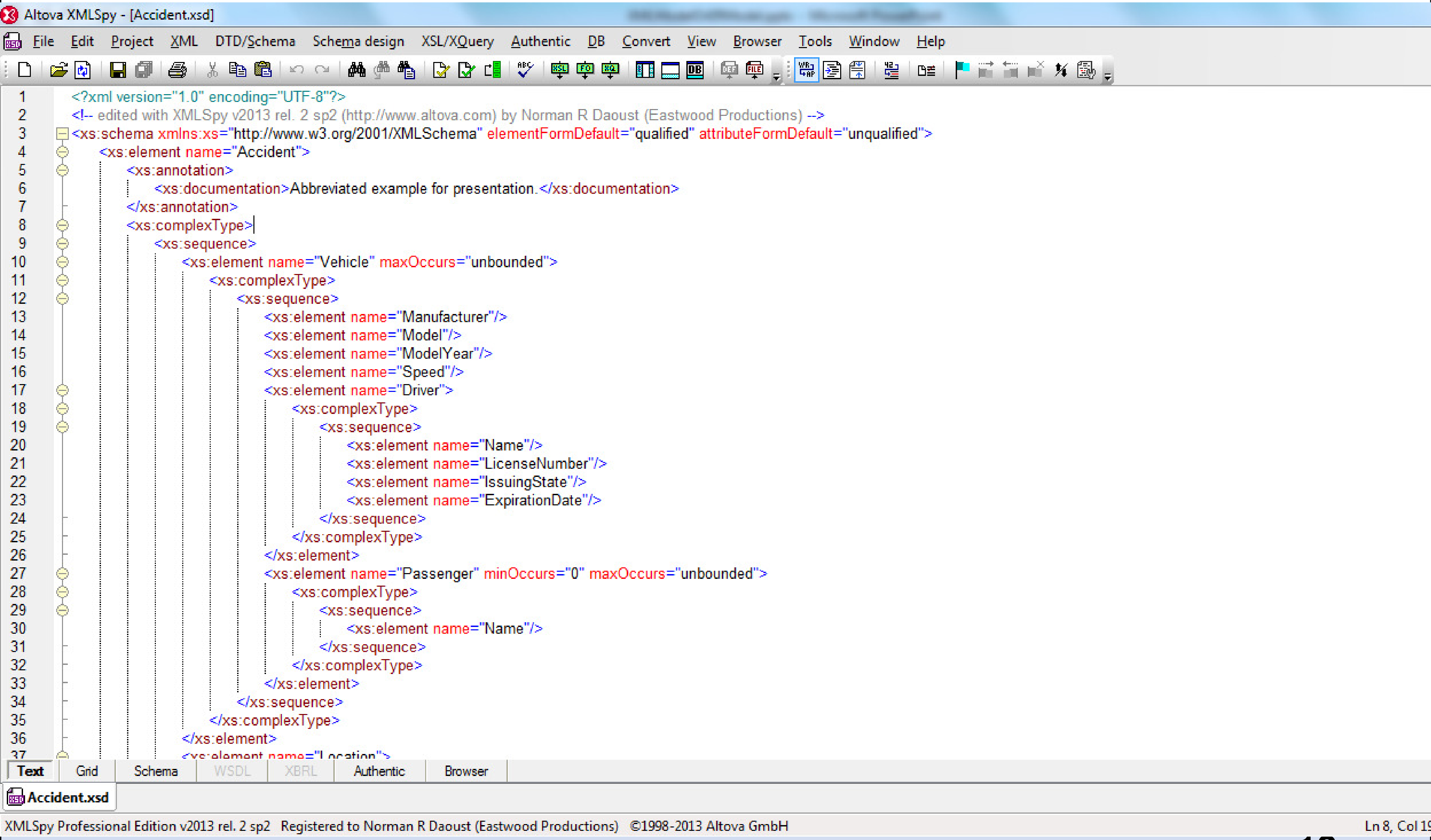


Altova XMLSpy - [Accident.xsd]
File Edit Project XML DTD/Schema Schema design XSL/XQuery Authentic DB Convert View Browser Tools Window Help

```
Comment  edited with XMLSpy v2013 rel. 2 sp2 (http://www.altova.com) by Norman R Daoust (Eastwood Productions)
xs:schema
    = xmlns:xs        http://www.w3.org/2001/XMLSchema
    = elementF...     qualified
    = attributeF...   unqualified
    xs:element
        = name        Accident
        xs:annotation
        xs:complexType
            xs:sequence
                xs:element
                    = name        Vehicle
                    = maxOccurs   unbounded
                    xs:complexType
                        xs:sequence
                            xs:element name=Manufacturer
                            xs:element name=Model
                            xs:element name=ModelYear
                            xs:element name=Speed
                            xs:element
                                = name        Driver
                                xs:complexType
                            xs:element
                                = name        Passenger
                                = minOccurs   0
                                = maxOccurs   unbounded
                                xs:complexType
                xs:element name=Datetime
                xs:element
                    = name        Location
                    xs:complexType
                        xs:sequence
                xs:element name=WeatherCondition
```

Text Grid Schema WSDL XBRL Authentic Browser
Accident.xsd
XMLSpy Professional Edition v2013 rel. 2 sp2 Registered to Norman R Daoust (Eastwood Productions) ©1998-2013 Altova GmbH

XML model: text representation, formatted (portion)


Altova XMLSpy - [Accident.xsd]
File Edit Project XML DTD/Schema Schema design XSL/XQuery Authentic DB Convert View Browser Tools Window Help
Text Grid Schema WSDL XBRL Authentic Browser
Accident.xsd
Ln 8, Col 19

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2013 rel. 2 sp2 (http://www.altova.com) by Norman R Daoust (Eastwood Productions) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:element name="Accident">
        <xs:annotation>
            <xs:documentation>Abbreviated example for presentation.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Vehicle" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Manufacturer"/>
                            <xs:element name="Model"/>
                            <xs:element name="ModelYear"/>
                            <xs:element name="Speed"/>
                            <xs:element name="Driver">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="Name"/>
                                        <xs:element name="LicenseNumber"/>
                                        <xs:element name="IssuingState"/>
                                        <xs:element name="ExpirationDate"/>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="Passenger" minOccurs="0" maxOccurs="unbounded">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="Name"/>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Location">
```

XMLSpy Professional Edition v2013 rel. 2 sp2 Registered to Norman R Daoust (Eastwood Productions) ©1998-2013 Altova GmbH

XML model: text representation (1 of 7)

```
<xs:element name="Accident">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Vehicle" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Manufacturer"/>
            <xs:element name="Model"/>
            <xs:element name="ModelYear"/>
            <xs:element name="Speed"/>
```

XML model: text representation (2 of 7)

```
<xs:element name="Driver"> <!-- child of Vehicle -->
 <xs:complexType>
  <xs:sequence>
   <xs:element name="Name"/>
   <xs:element name="LicenseNumber"/>
   <xs:element name="IssuingState"/>
   <xs:element name="ExpirationDate"/>
  </xs:sequence>
 </xs:complexType>
</xs:element>
```

XML model: text representation (3 of 7)

```
<!-- child of Vehicle -->

<xs:element name="Passenger" minOccurs="0" maxOccurs="unbounded">

   <xs:complexType>

        <xs:sequence>

            <xs:element name="Name"/>

        </xs:sequence>

    </xs:complexType>

</xs:element>
```

XML model: text representation (4 of 7)

```
    </xs:sequence>
  </xs:complexType>
</xs:element> <!-- end of Vehicle -->
```

XML model: text representation (5 of 7)

```
<xs:element name="Location">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PlaceSpecification"/>
      <xs:element name="City"/>
      <xs:element name="StateCode"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

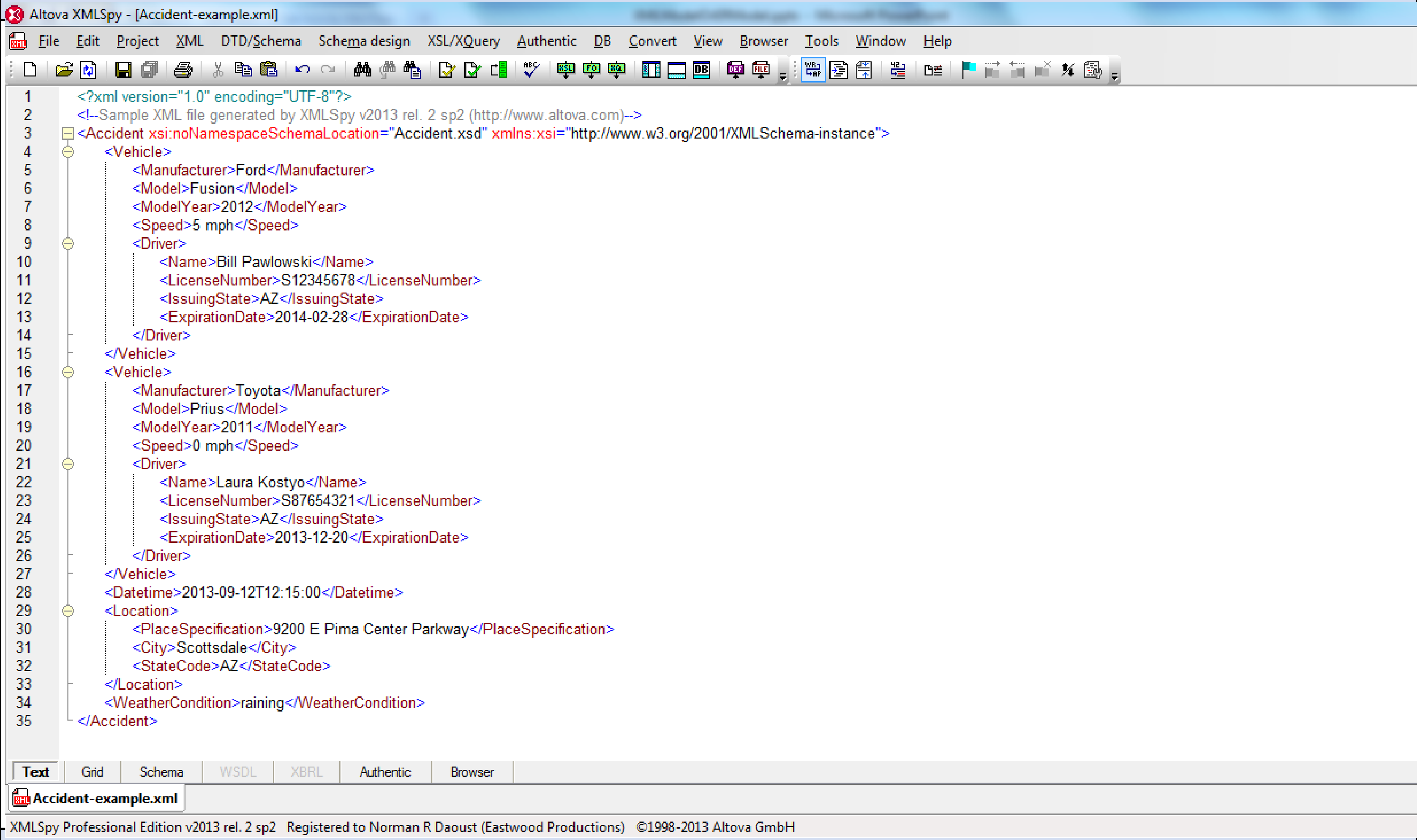
XML model: text representation (6 of 7)

```
<xs:element name="RoadCondition">
  <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="clear"/>
        <xs:enumeration value="raining"/>
        <xs:enumeration value="snowing"/>
     </xs:restriction>
   </xs:simpleType>
</xs:element>
```

XML model: text representation (7 of 7)

```
    </xs:sequence>
  </xs:complexType>
 </xs:element> <! end of Accident -->
</xs:schema>
```

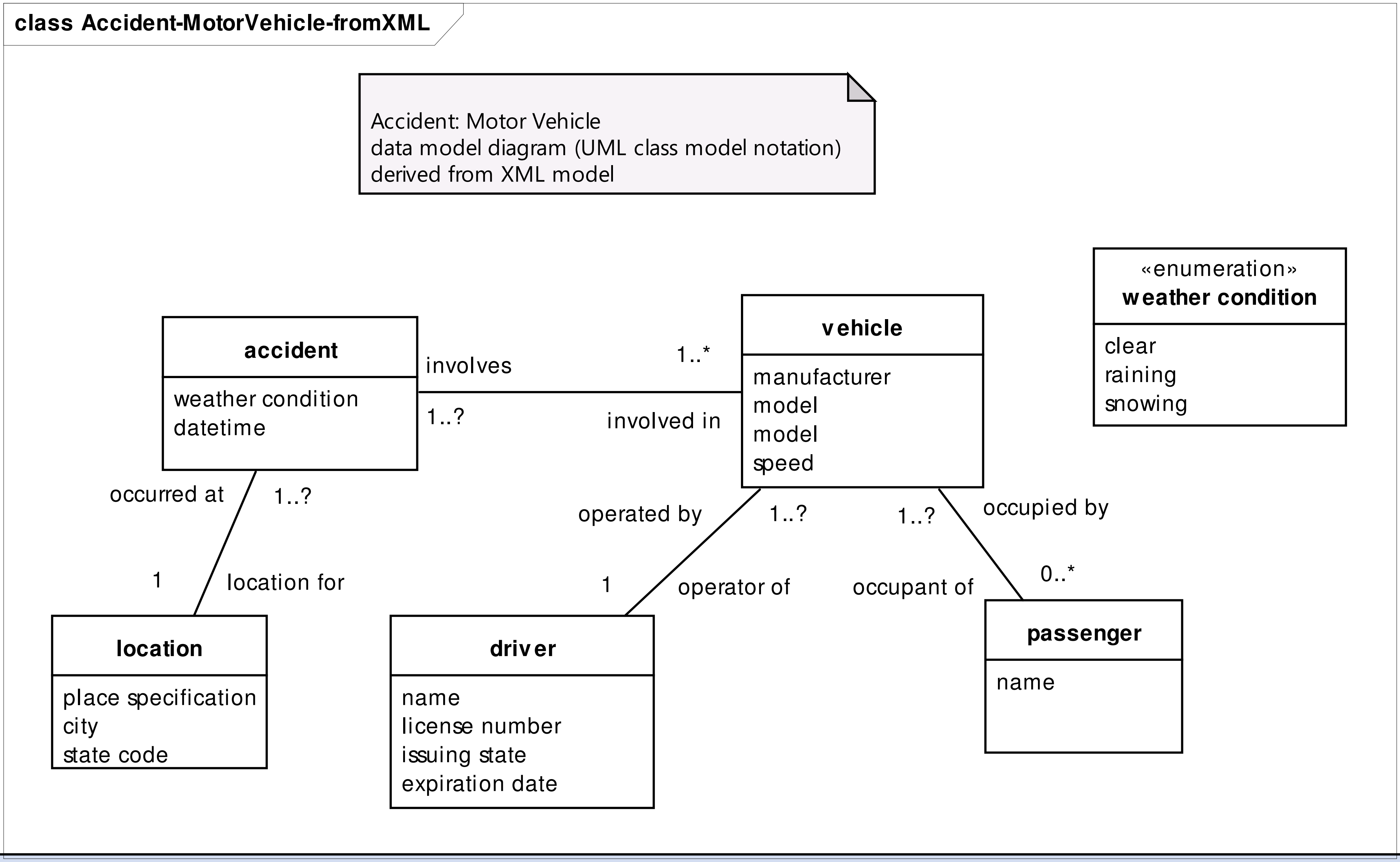
XML model: instance (XML document)


Altova XMLSpy - [Accident-example.xml]
File Edit Project XML DTD/Schema Schema design XSL/XQuery Authentic DB Convert View Browser Tools Window Help
Text Grid Schema WSDL XBRL Authentic Browser
Accident-example.xml

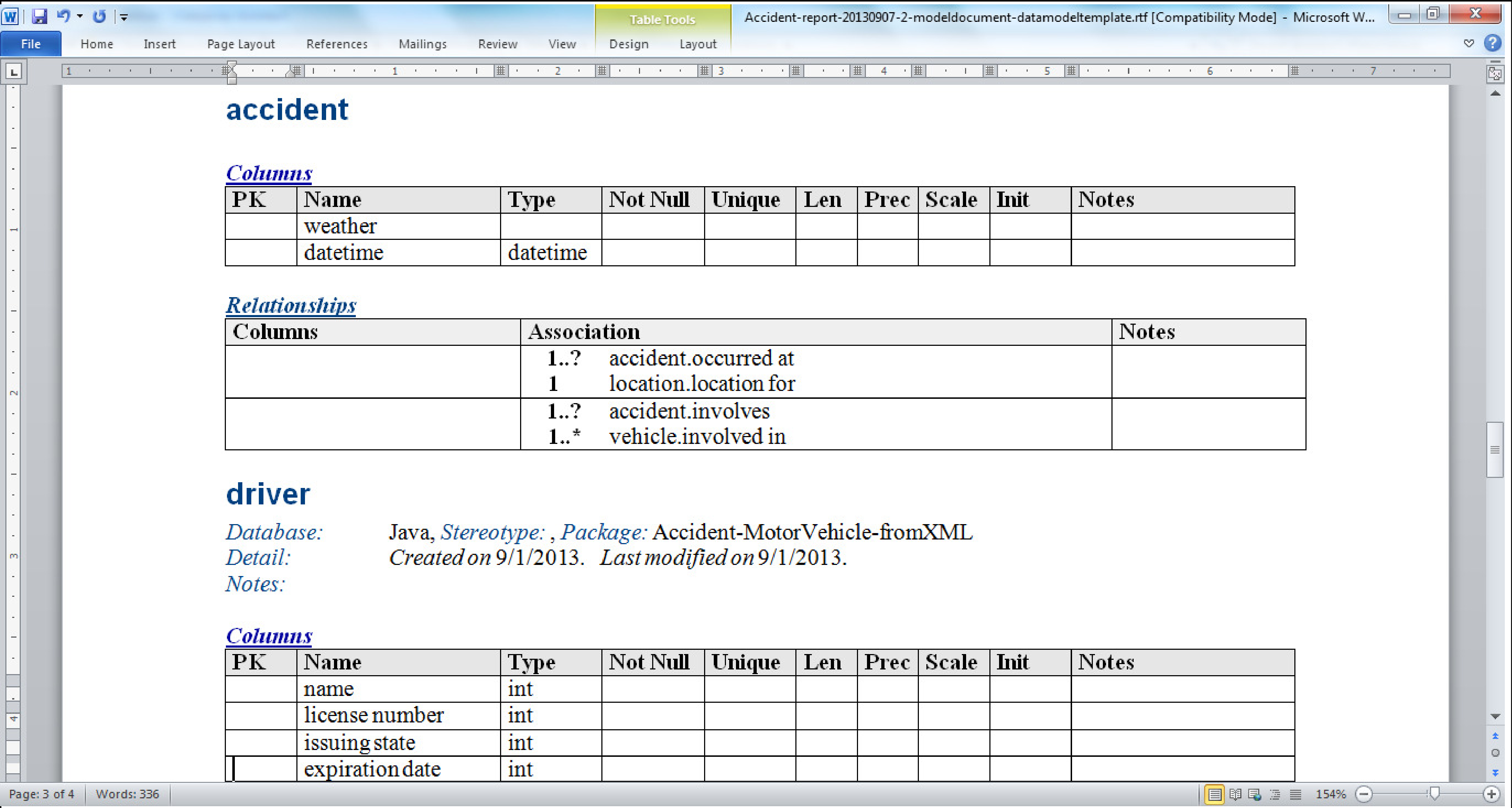
```xml
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2013 rel. 2 sp2 (http://www.altova.com)-->
<Accident xsi:noNamespaceSchemaLocation="Accident.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Vehicle>
        <Manufacturer>Ford</Manufacturer>
        <Model>Fusion</Model>
        <ModelYear>2012</ModelYear>
        <Speed>5 mph</Speed>
        <Driver>
            <Name>Bill Pawlowski</Name>
            <LicenseNumber>S12345678</LicenseNumber>
            <IssuingState>AZ</IssuingState>
            <ExpirationDate>2014-02-28</ExpirationDate>
        </Driver>
    </Vehicle>
    <Vehicle>
        <Manufacturer>Toyota</Manufacturer>
        <Model>Prius</Model>
        <ModelYear>2011</ModelYear>
        <Speed>0 mph</Speed>
        <Driver>
            <Name>Laura Kostyo</Name>
            <LicenseNumber>S87654321</LicenseNumber>
            <IssuingState>AZ</IssuingState>
            <ExpirationDate>2013-12-20</ExpirationDate>
        </Driver>
    </Vehicle>
    <Datetime>2013-09-12T12:15:00</Datetime>
    <Location>
        <PlaceSpecification>9200 E Pima Center Parkway</PlaceSpecification>
        <City>Scottsdale</City>
        <StateCode>AZ</StateCode>
    </Location>
    <WeatherCondition>raining</WeatherCondition>
</Accident>
```

XMLSpy Professional Edition v2013 rel. 2 sp2   Registered to Norman R Daoust (Eastwood Productions)   ©1998-2013 Altova GmbH

ER model: derived from XML model



ER model: text representation (portion)



accident

Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
	weather								
	datetime	datetime							

Relationships

Columns	Association	Notes
	1..? accident.occurred at 1 location.location for	
	1..? accident.involves 1..* vehicle.involved in	

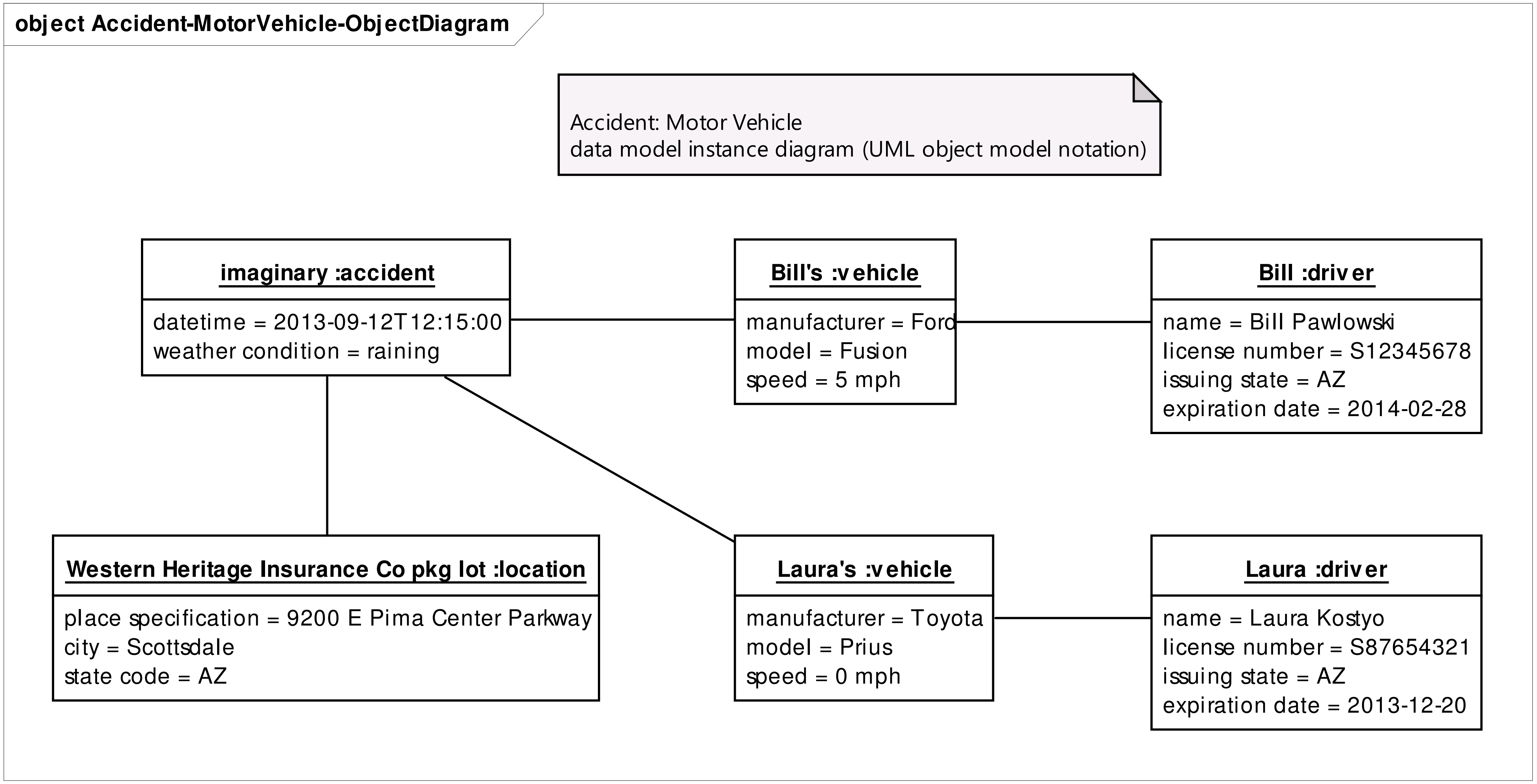
driver

Database: Java, *Stereotype:* , *Package:* Accident-MotorVehicle-fromXML
Detail: *Created on* 9/1/2013. *Last modified on* 9/1/2013.
Notes:

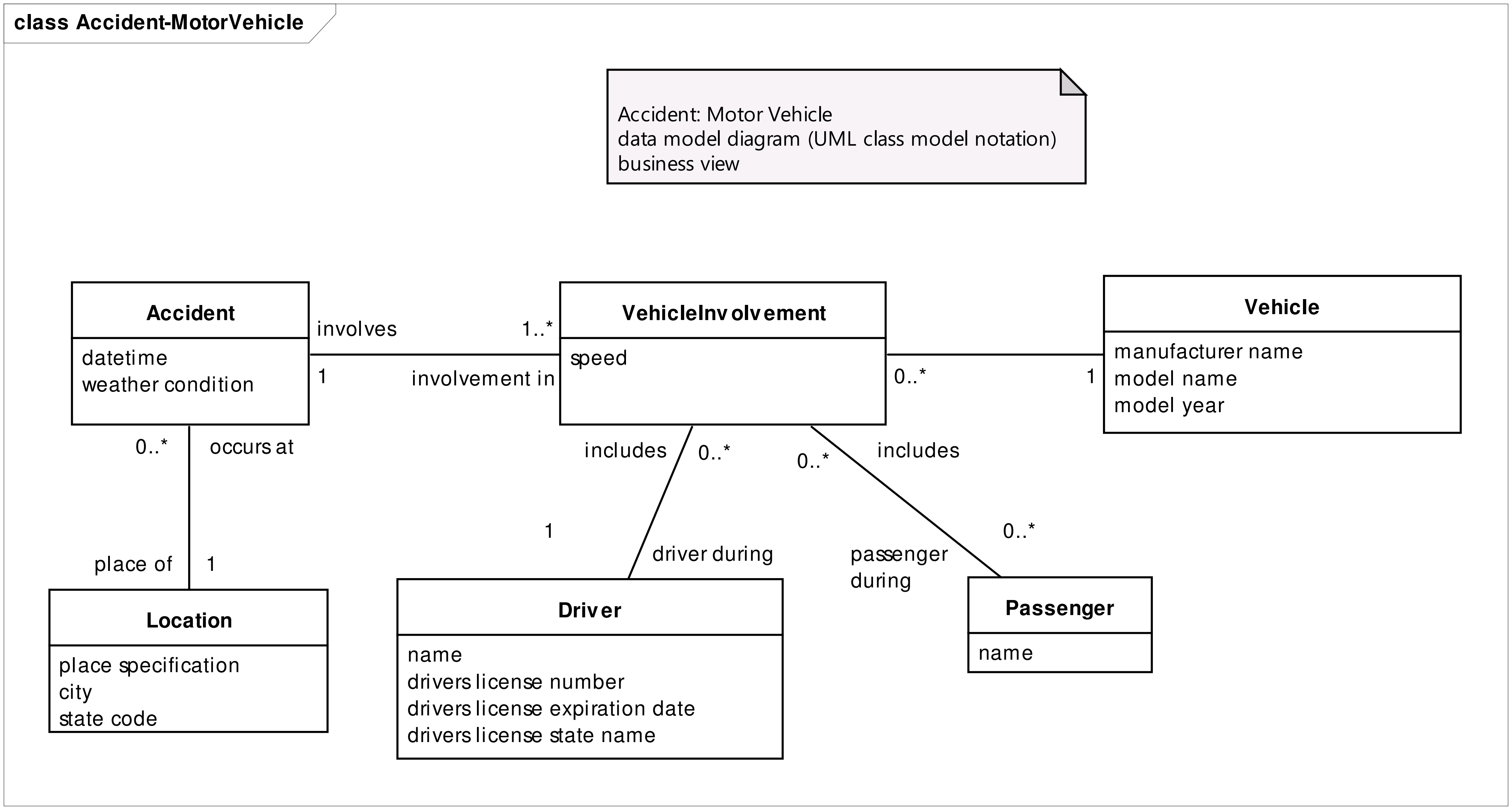
Columns

PK	Name	Type	Not Null	Unique	Len	Prec	Scale	Init	Notes
	name	int							
	license number	int							
	issuing state	int							
	expiration date	int							

ER model: instance diagram


object Accident-MotorVehicle-ObjectDiagram
Accident: Motor Vehicle
data model instance diagram (UML object model notation)
imaginary :accident
datetime = 2013-09-12T12:15:00
weather condition = raining
Bill's :vehicle
manufacturer = Ford
model = Fusion
speed = 5 mph
Bill :driver
name = Bill Pawlowski
license number = S12345678
issuing state = AZ
expiration date = 2014-02-28
Western Heritage Insurance Co pkg lot :location
place specification = 9200 E Pima Center Parkway
city = Scottsdale
state code = AZ
Laura's :vehicle
manufacturer = Toyota
model = Prius
speed = 0 mph
Laura :driver
name = Laura Kostyo
license number = S87654321
issuing state = AZ
expiration date = 2013-12-20

ER model: not derived from XML model



Comparisons: XML vs. ER Model

- graphical format
- text format
- target audience
- datatypes
- adoption
- acceptance
- tool support
- advantages and disadvantages
- criteria for selection
- appropriateness for database, message, and services design

graphical format: XML vs. ER Model

ER	XML
A/D: several relatively standard formats: Entity Relationship (ER)/Barker Notation, IDEF1X Notation, UML (primary and foreign keys illustrated in different ways in different tools)	D: each vendor is different
A: ability to illustrate relationships in both directions	D: no ability to illustrate relationships in both directions
	D: hierarchical only

text format: XML vs. ER Model

ER	XML
D: each tool vendor may have their own default format	A: standard XML schema definition from W3C industry standards organization

target audience: XML vs. ER Model

ER	XML
data modelers, DBAs, software developers, business subject matter experts	software developers, business subject matter experts

datatypes: XML vs. ER Model

ER	XML
D: no standard datatypes (except DBMS-specific for physical data models; ISO/IEC 11404, Information technology — General-Purpose Datatypes (GPD),for programming languages and software interfaces, seldom used in data models)	A: standard XML schema datatypes

adoption: XML vs. ER Model

ER	XML
D: niche, only by data modelers	A: widespread, becoming ubiquitous

acceptance: XML vs. ER Model

ER	XML
widespread in data management community, not widely used in software development community	widespread, accepted even when perhaps not the most appropriate approach

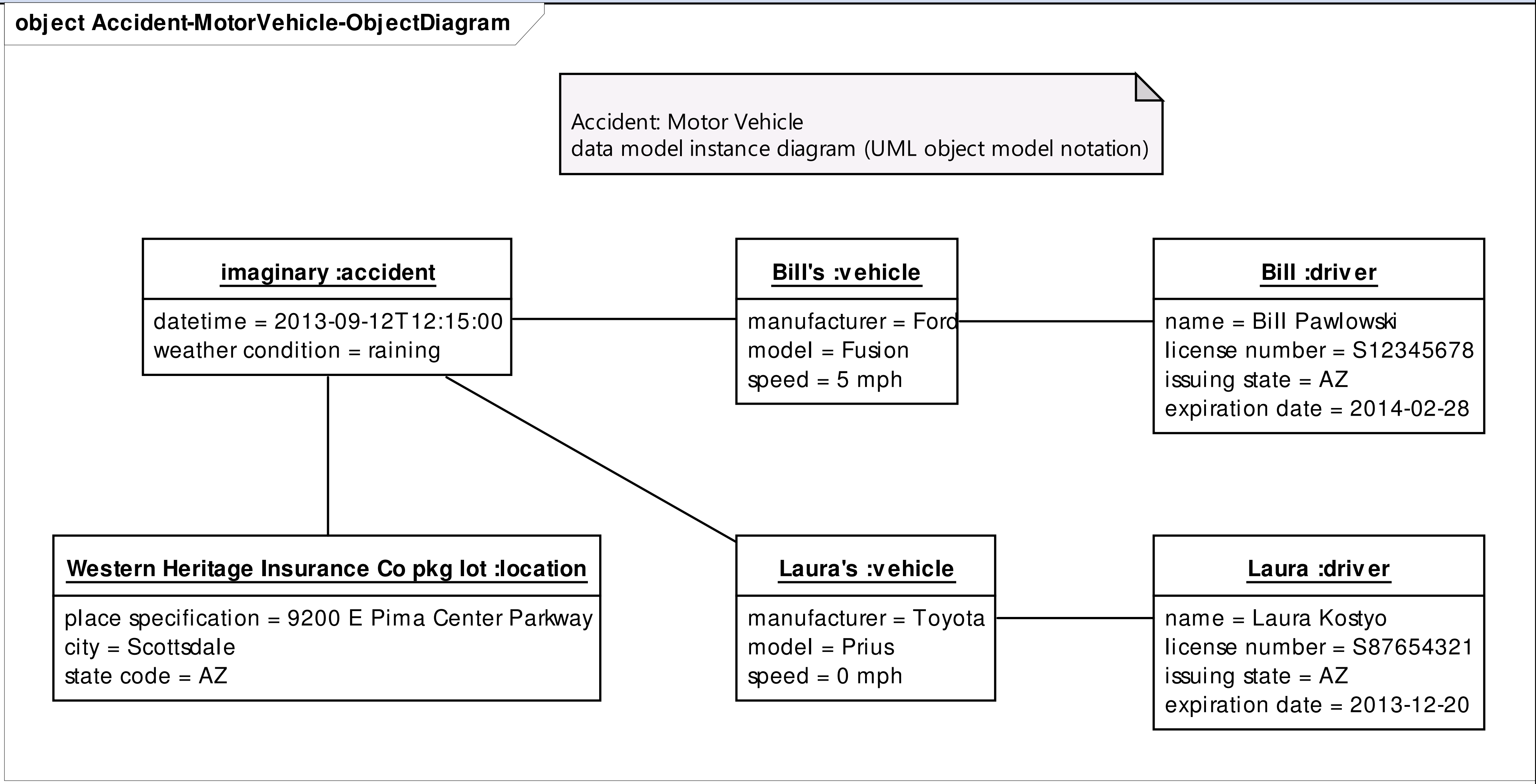
tool support: XML vs. ER Model

ER	XML
Diagram Creation: D: limited	Diagram Creation: D: limited
Text Creation: D: limited/not applicable since no standard format	Text Creation: A: widespread (any text editor)

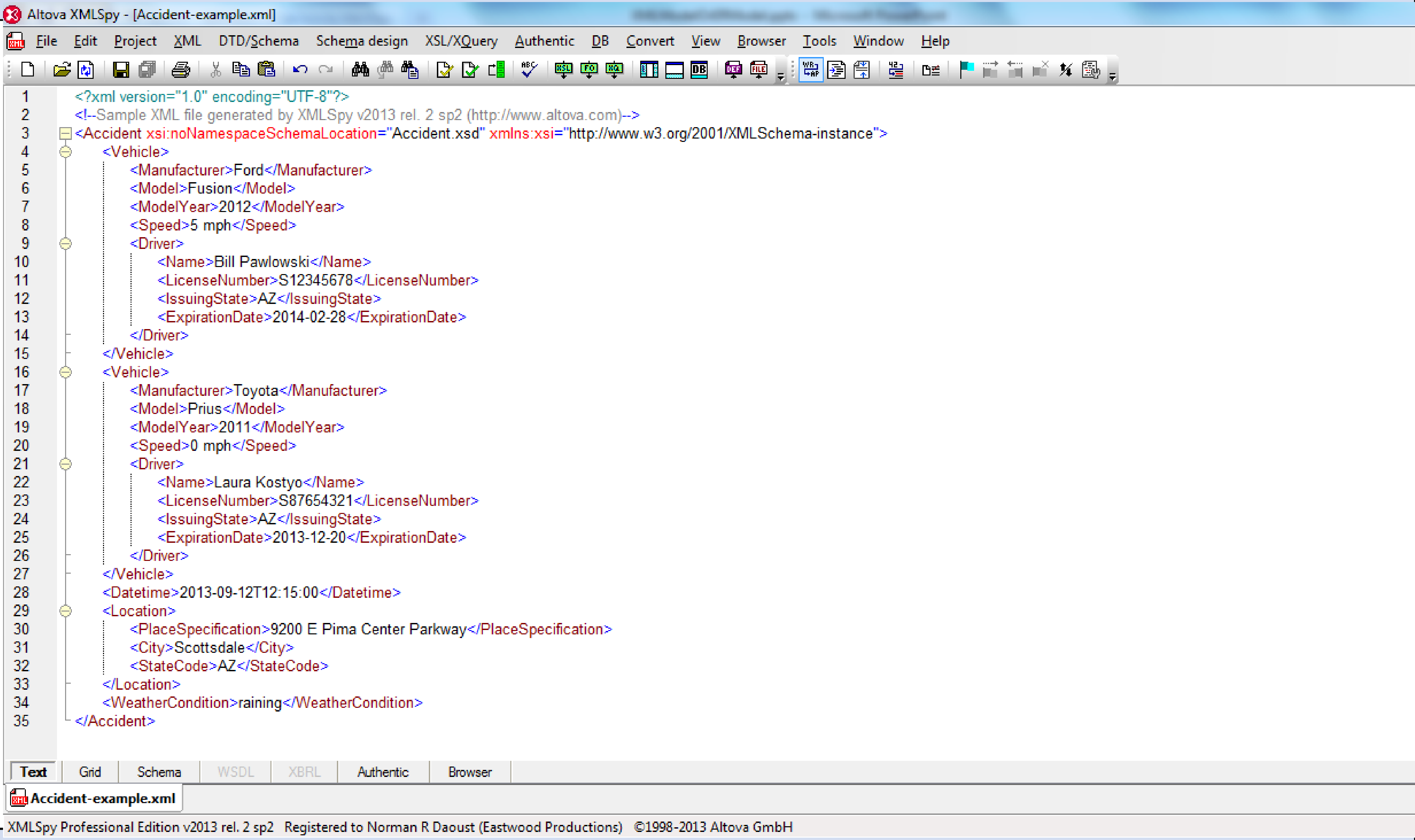
advantages and disadvantages: XML vs. ER Model

ER	XML
A: normalization rules	
D: no standard format for representing instances/data contents (except UML object diagrams) (spreadsheets, text are non-standard)	A: widespread tool support for reading XML instance formats (every web browser and text editor can display an XML document)
D: no specified serialization format for instances (although one can be derived from the model)	A: defines serialization format in XML for instances and electronic data exchange
D: no standard datatypes	A: standard datatypes in XML schema

instance example: ER (UML object diagram)


object Accident-MotorVehicle-ObjectDiagram
Accident: Motor Vehicle
data model instance diagram (UML object model notation)
imaginary :accident
datetime = 2013-09-12T12:15:00
weather condition = raining
Bill's :vehicle
manufacturer = Ford
model = Fusion
speed = 5 mph
Bill :driver
name = Bill Pawlowski
license number = S12345678
issuing state = AZ
expiration date = 2014-02-28
Western Heritage Insurance Co pkg lot :location
place specification = 9200 E Pima Center Parkway
city = Scottsdale
state code = AZ
Laura's :vehicle
manufacturer = Toyota
model = Prius
speed = 0 mph
Laura :driver
name = Laura Kostyo
license number = S87654321
issuing state = AZ
expiration date = 2013-12-20

instance example: XML document


Altova XMLSpy - [Accident-example.xml]
File Edit Project XML DTD/Schema Schema design XSL/XQuery Authentic DB Convert View Browser Tools Window Help
Text Grid Schema WSDL XBRL Authentic Browser
Accident-example.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2013 rel. 2 sp2 (http://www.altova.com)-->
<Accident xsi:noNamespaceSchemaLocation="Accident.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Vehicle>
        <Manufacturer>Ford</Manufacturer>
        <Model>Fusion</Model>
        <ModelYear>2012</ModelYear>
        <Speed>5 mph</Speed>
        <Driver>
            <Name>Bill Pawlowski</Name>
            <LicenseNumber>S12345678</LicenseNumber>
            <IssuingState>AZ</IssuingState>
            <ExpirationDate>2014-02-28</ExpirationDate>
        </Driver>
    </Vehicle>
    <Vehicle>
        <Manufacturer>Toyota</Manufacturer>
        <Model>Prius</Model>
        <ModelYear>2011</ModelYear>
        <Speed>0 mph</Speed>
        <Driver>
            <Name>Laura Kostyo</Name>
            <LicenseNumber>S87654321</LicenseNumber>
            <IssuingState>AZ</IssuingState>
            <ExpirationDate>2013-12-20</ExpirationDate>
        </Driver>
    </Vehicle>
    <Datetime>2013-09-12T12:15:00</Datetime>
    <Location>
        <PlaceSpecification>9200 E Pima Center Parkway</PlaceSpecification>
        <City>Scottsdale</City>
        <StateCode>AZ</StateCode>
    </Location>
    <WeatherCondition>raining</WeatherCondition>
</Accident>
```

XMLSpy Professional Edition v2013 rel. 2 sp2   Registered to Norman R Daoust (Eastwood Productions)   ©1998-2013 Altova GmbH

Appropriateness

- for database design
- for message design
- for services

appropriateness for database design

- XML
  - A: appropriate only for design of XML databases
- ER
  - A: appropriate for design of relational databases, less appropriate for design of XML databases

appropriateness for message design

- XML
  - A: appropriate for design of messages in XML format
- ER
  - D: not directly appropriate, but message design can be systematically derived from an ER model

appropriateness for services

- XML
  - A: appropriate for specification of input, output, error messages of each operation (assuming XML message format)
- ER
  - D: not appropriate for specification of input, output, error messages of each operation, however their specification can be derived from an ER model

criteria for selection: XML or ER Model

- ➢If goal is relational database design, ER model is best
- ➢If goal is XML database design, XML model is best, ER model is possible
- ➢If goal is message design, XML model is appropriate for XML messages, message design can be systematically derived from an ER model
- ➢If goal is determining a service and its operations, neither is appropriate
- ➢If goal is defining operation input and output messages for a service, XML model is appropriate for XML messages, message design can be systematically derived from an ER model

Deriving XML Message Format from ER Model

1. determine message category (e.g., request/response, query/query response, notification) and triggering event (if applicable) and message type (message structure)

2. determine starting entity (typically explicit in business purpose, message type, or triggering event)

3. select applicable attributes of the entity

4. sequentially select all applicable associations and their target entities and attributes (and sequentially select those entities’ applicable associations and target entity and attributes, and …)

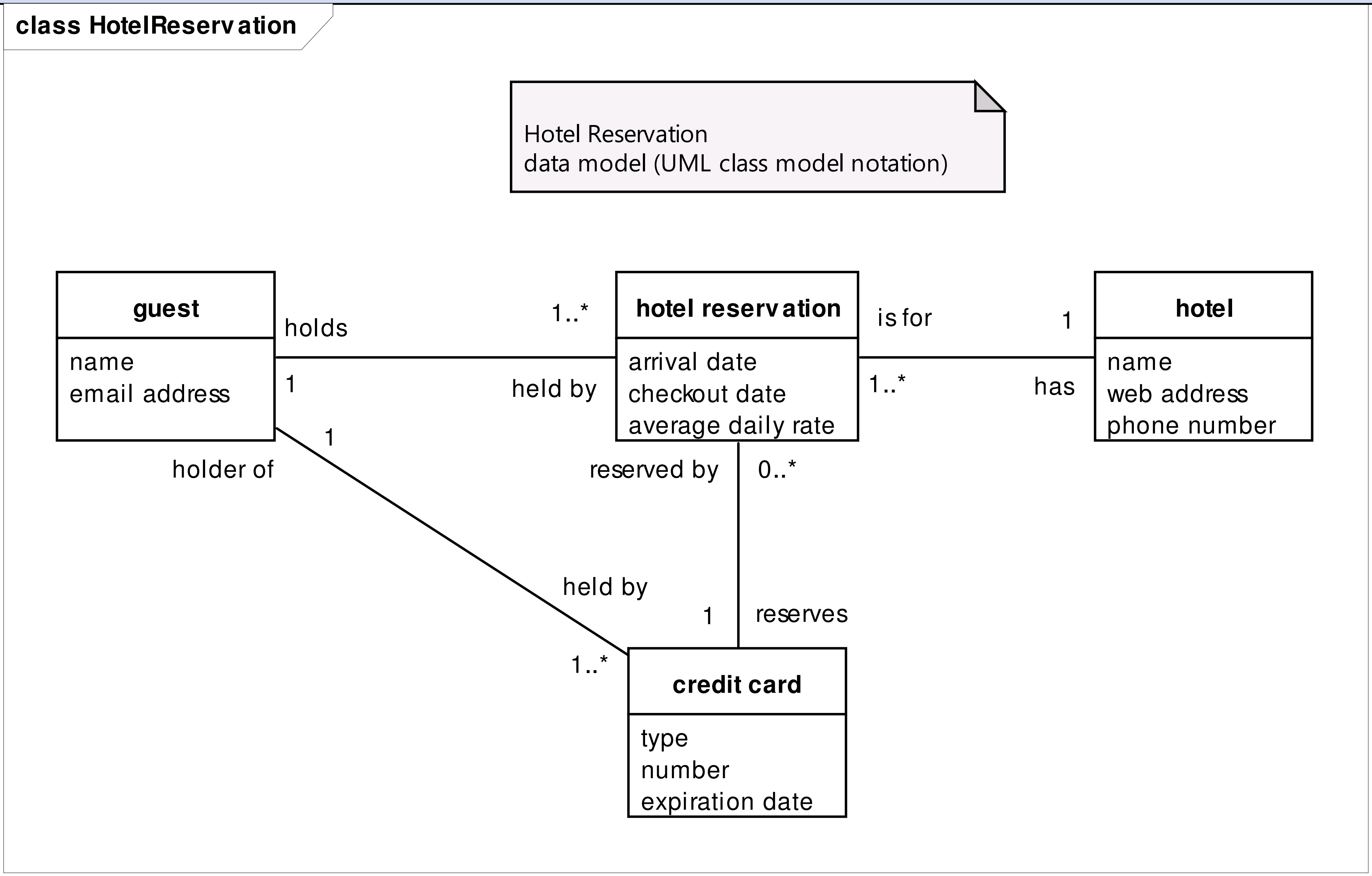
Derive from ER model: example

- list of future hotel reservations for a guest

Derive from ER model, process: 1 of 7

- 1. determine message category (e.g., request/response, query/query response, notification) and triggering event (if applicable) and message type (message structure)
- message category: **query response**
- triggering event: **not applicable**
- message type: **future hotel reservations for a guest**

Derive from ER model: ER model



Derive from ER model, process: model path

- guest (A)
  - hotel reservation (A.1)
    - hotel (A.1.a)
    - credit card (A.1.b)
      - (guest: no) (A.1.b.1)
  - (credit card: no) (A2)

Derive from ER model, process: 2 of 7

- 2. determine starting entity (typically explicit in business purpose, message type, or triggering event)
- starting entity: guest (A)

Derive from ER model, process: 3 of 7

- 3. select applicable attributes of the entity
- attributes of <u>guest</u>: name, email address (A)

Derive from ER model, process: 4 of 7

- 4. sequentially select all applicable associations and their target entities and attributes (and sequentially select those entities' applicable associations and target entity and attributes, and …)
- first association from guest: holds one-to-many hotel reservation
- target entity: hotel reservation (A.1)
- target entity attributes: arrival date, checkout date, average daily rate
- target entity (hotel reservation) first association: is for one hotel (A.1.a)

Derive from ER model, process: 5 of 7

- target entity (hotel reservation) first association: is for one <u>hotel</u> (A.1.a)
- target entity: <u>hotel</u> (A.1.a)
- target entity attributes: name, phone number, web address
- target entity (hotel) first association: **no additional associations**

Derive from ER model, process: 6 of 7

- target entity (hotel reservation) second association: reserved by one <u>credit</u> <u>card</u>

- target entity: <u>credit</u> <u>card</u> (A.1.b)

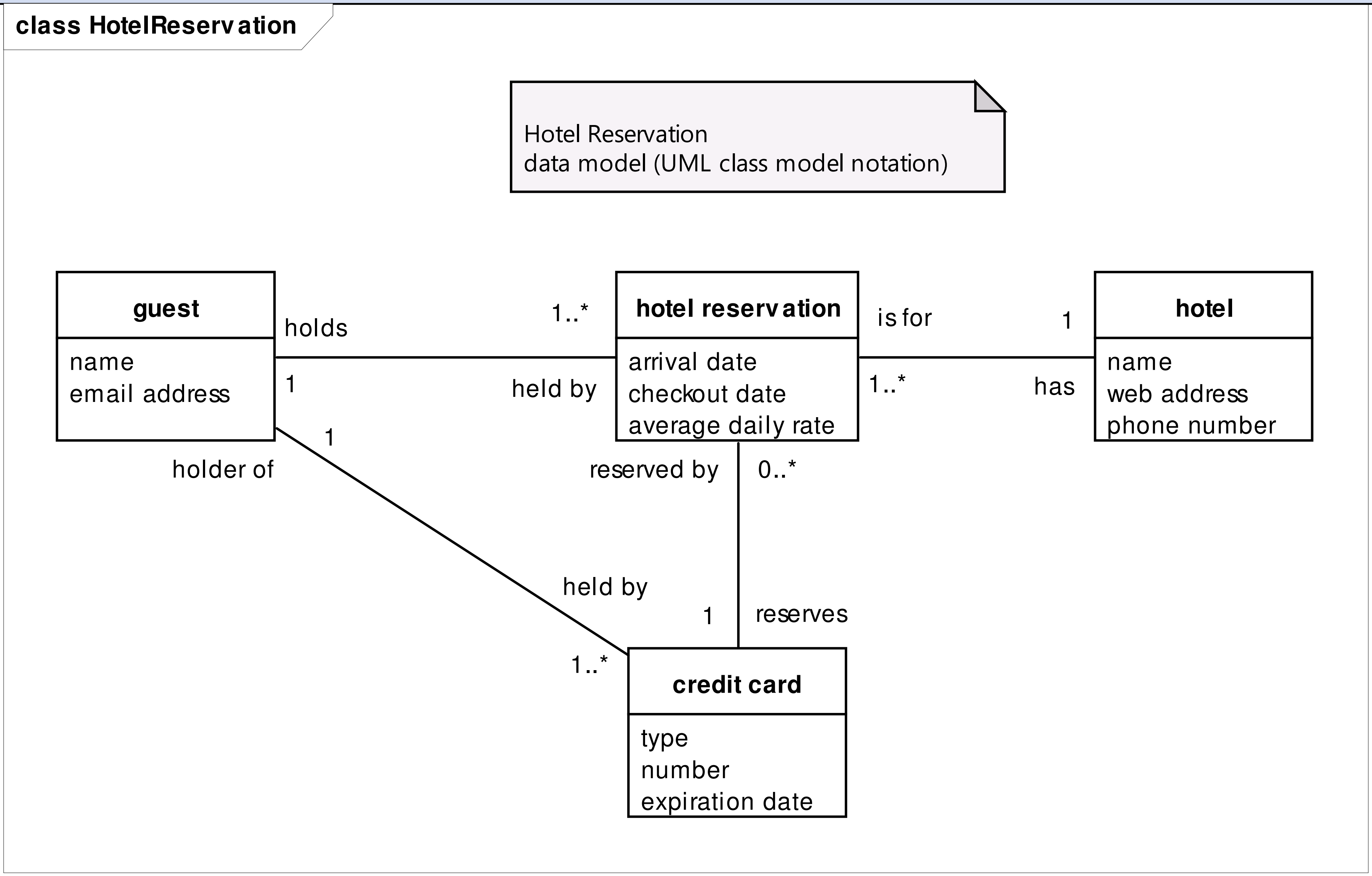
- target entity attributes: number **(other attributes not applicable)**

- target entity (<u>credit</u> <u>card</u>) first association: <u>guest</u> (A.1.b.1): **not applicable, don't select the association**

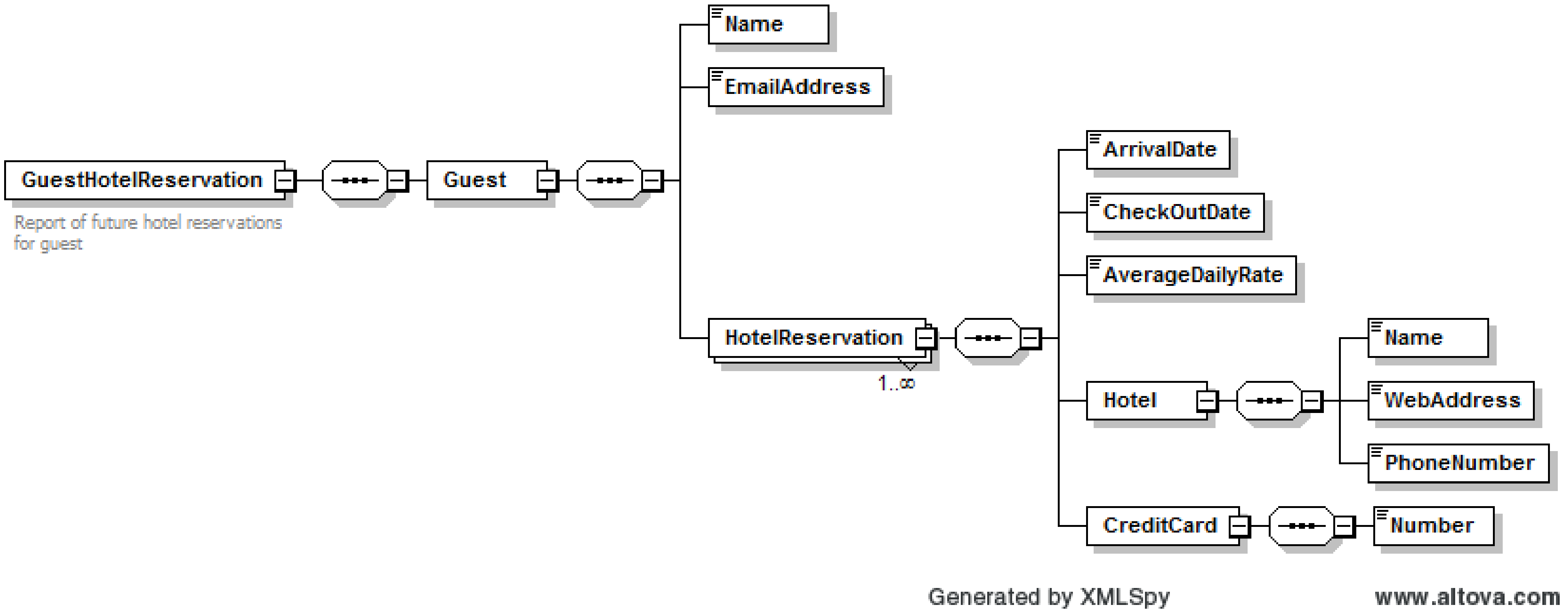
Derive from ER model, process: 7 of 7

- second association from guest: holder of one-to-many credit card (A.2): **not applicable, don’t select the association**

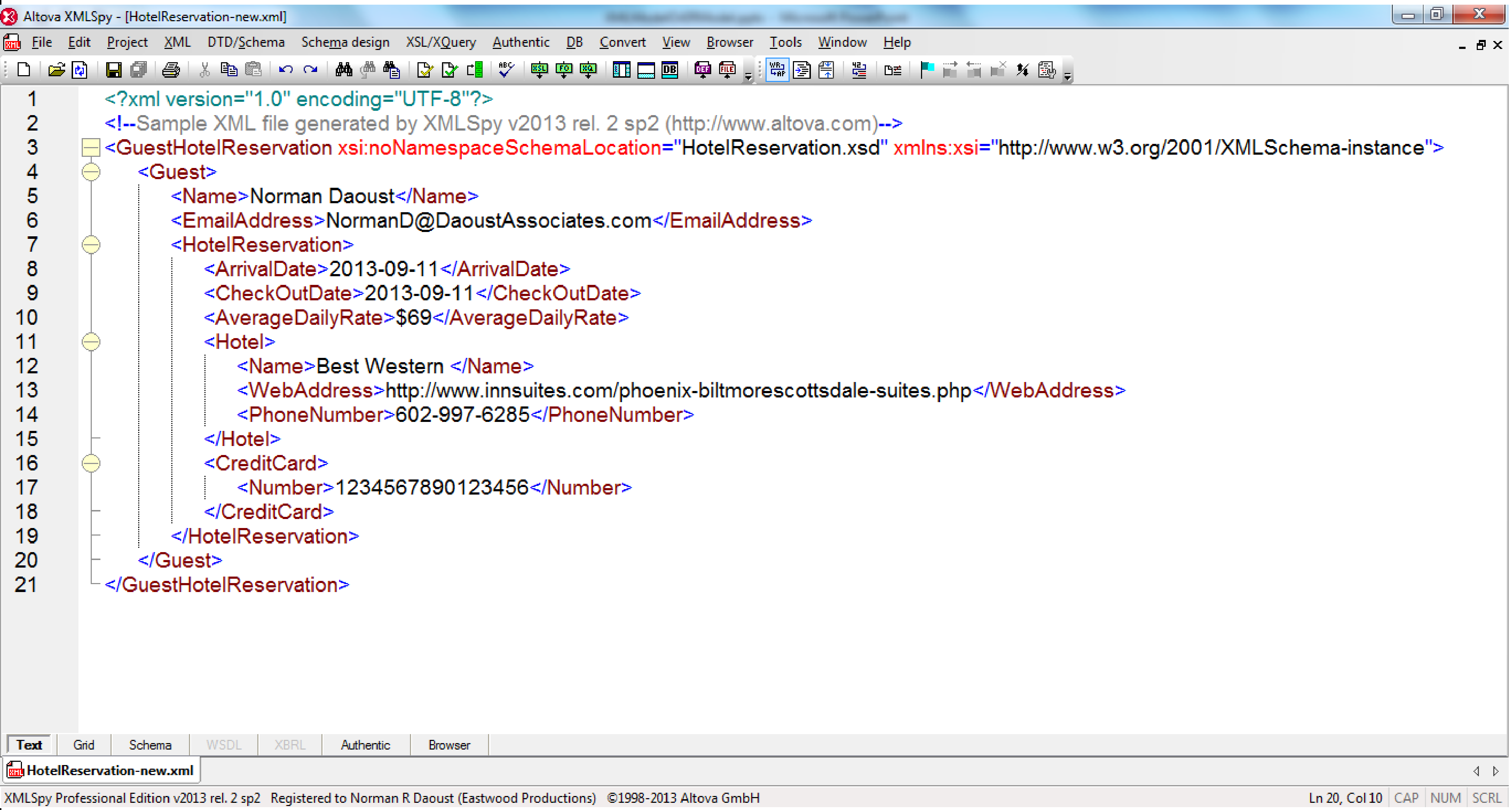
Derive from ER model: ER model



XML schema from ER model



XML document from XML schema from ER model



Altova XMLSpy - [HotelReservation-new.xml]
File Edit Project XML DTD/Schema Schema design XSL/XQuery Authentic DB Convert View Browser Tools Window Help
Text Grid Schema WSDL XBRL Authentic Browser
HotelReservation-new.xml
XMLSpy Professional Edition v2013 rel. 2 sp2 Registered to Norman R Daoust (Eastwood Productions) ©1998-2013 Altova GmbH
Ln 20, Col 10 CAP NUM SCRL

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2013 rel. 2 sp2 (http://www.altova.com)-->
<GuestHotelReservation xsi:noNamespaceSchemaLocation="HotelReservation.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Guest>
        <Name>Norman Daoust</Name>
        <EmailAddress>NormanD@DaoustAssociates.com</EmailAddress>
        <HotelReservation>
            <ArrivalDate>2013-09-11</ArrivalDate>
            <CheckOutDate>2013-09-11</CheckOutDate>
            <AverageDailyRate>$69</AverageDailyRate>
            <Hotel>
                <Name>Best Western </Name>
                <WebAddress>http://www.innsuites.com/phoenix-biltmorescottsdale-suites.php</WebAddress>
                <PhoneNumber>602-997-6285</PhoneNumber>
            </Hotel>
            <CreditCard>
                <Number>1234567890123456</Number>
            </CreditCard>
        </HotelReservation>
    </Guest>
</GuestHotelReservation>
```

Advantages of deriving XML models/messages from ER model

- insures holistic view by looking at relationships in both directions, not just one; that may eliminate analysis blind spots
- typically results in increased consistency and reuse
- typically results in higher quality XML models

Disadvantages of deriving XML models/messages from ER model

- takes more time
- you’ll ask questions that people may not have thought about and don’t have answers to
- requires thinking about possible future uses

References

- www.w3.org/TR/#tr_XML_Schema, XML schema specification
- en.wikipedia.org/wiki/XML_Schema_Editor, XML schema tools
- en.wikipedia.org/wiki/Comparison_of_XML_editors, XML schema tools comparison
- http://en.wikipedia.org/wiki/Comparison_of_data_modeling_tools, data modeling tools
- *SOA Principles of Service Design*, Prentice Hall, 2008, Thomas Erl
- *Succeeding with SOA*, Addison Wesley, 2007, Paul C. Brown
- *UML Requirements Modeling for Business Analysts*, Technics Publications, 2012, Norman Daoust

Summary

- To understand a business or business area, create an ER model
- To design a relational database to provide the data for a business or business area, create an ER model
- To design an XML data exchange format, create an XML model, preferably derived from an ER model
- To design a set of consistent data exchange formats across the enterprise, derive the data exchange formats from an ER model

Presentation Outline

- Why should you care?
- Definitions
- Examples
- Comparison: XML model vs.. entity relationship model
- Deriving XML models from an entity relationship model
- References
- Summary

Presentation Goals

- similarities and differences between entity relationship data modeling and XML modeling
- advantages and disadvantages of both entity relationship data models and XML models
- appropriate usages for data models and XML models
- criteria for determining which to create

Quote

- “To a person with a hammer, every problem looks like a nail.”
- Daoust Associates corollary: “A modeler using multiple tools can create useful and high quality deliverables for their organization.”

Thanks!

Norman Daoust

Daoust Associates

www.daoustassociates.com

NormanD@DaoustAssociates.com

(617) 491-7424