Auto Insurance Eligibility Model

Domain Model

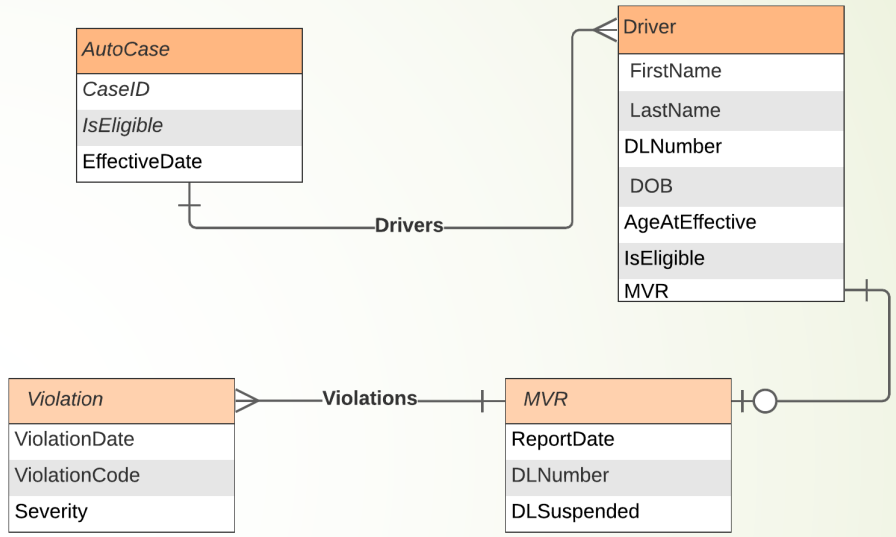
- Auto Insurance Case
- Drivers
- Motor Vehicle Reports (MVR)
- Violations

Relations:

- Each AutoCase has one or more drivers.
- Each Driver has one MVR record.
- Each MVR record has a collection of violations.

Decision Logic

- For case to be eligible, all drivers must be eligible.
- For a driver to be eligible:
  - Driver must be over 18 years of age.
  - Driver's MVR must contain
    - 5 or less "Major" violations in total.
    - 2 or less "Major" violations in the last year.
  - Driver's MVR must not show current suspension.
- Driver's age = Difference in years between policy effective date and driver's birthdate.



Domain Model

```
<model>
  <object-type name="AutoCase@eg.maximal.co" label="Auto Insurance Case">
    <attribute name="CaseID" label="Id of the case." datatype="text" optional="false"/>
    <attribute name="EffectiveDate" label="Policy effective date" datatype="date"/>
    <attribute name="IsEligible" label="Whether or not case is eligible" datatype="boolean"/>
    <relation name="Drivers" label="Drivers in the case" inverse-object="Driver@eg.maximal.co"
      inverse-attribute="AutoCase" is-containment="true"/>
    <persistence-params table-name="auto_case"/>
  </object-type>

  <object-type name="Driver@eg.maximal.co" label="Driver information.">
    <attribute name="AutoCase" label="The case driver belongs to." datatype="reference"
      ref-type="AutoCase@eg.maximal.co"/>
    <attribute name="FirstName" label="First name of the driver." datatype="text"/>
    <attribute name="LastName" label="Last name of the driver." datatype="text"/>
    <attribute name="DLNumber" label="Driver license number." datatype="text"/>
    <attribute name="DOB" label="Date of birth." datatype="date"/>
    <attribute name="AgeAtEffective" label="Age at policy effective date." datatype="integer"/>
    <attribute name="IsEligible" label="Whether or not driver is eligible" datatype="boolean"/>
    <attribute name="MVR" label="MVR for the driver" datatype="reference" ref-type="MVR@eg.maximal.co"/>
    <persistence-params table-name="auto_driver"/>
  </object-type>

  <object-type name="MVR@eg.maximal.co" label="Motor vehicle report.">
    <attribute name="ReportDate" label="Date of MVR report." datatype="date"/>
    <attribute name="DLSuspended" label="Is driver license suspended?" datatype="boolean"/>
    <relation name="Violations" label="Violations in the MVR" inverse-object="Violation@eg.maximal.co"
      inverse-attribute="MVR" is-containment="true"/>
    <persistence-params table-name="auto_mvr"/>
  </object-type>

   <object-type name="Violation@eg.maximal.co" label="Motor vehicle report violation.">
    <attribute name="MVR" label="The MVR the violation belongs to." datatype="reference"
      ref-type="MVR@eg.maximal.co"/>
    <attribute name="ViolationDate" label="Violation date." datatype="date"/>
    <attribute name="ViolationCode" label="Violation code" datatype="date"/>
    <attribute name="Severity" label="Violation severity" datatype="enumerated"
      enumtype="ViolationSeverityEnum@eg.maximal.co"/>
    <persistence-params table-name="auto_violation"/>
  </object-type>

  <enum-type name="ViolationSeverityEnum@eg.maximal.co" label="Violation Severity Violation">
    <enum-option name="MAJOR" label="Major"/>
    <enum-option name="MINOR" label="Minor"/>
  </enum-type>
</model>
```

"Drivers" is a 1-to-N relationship between AutoCase and Drivers. Inverse-object and inverse-attribute are the key to defining it.

Each driver has one MVR, defined as an attribute of "reference" data type.

"Violations" is a 1-N relationship between an MVR and Violations.

Decision Logic

maximal

Sets

- A "set" in Maximal is a collection of objects, defined either by the type of object or by a relation.
- When a set is referenced in a constraint, it means that constraint is applicable to all members of that set. It is a way of saying "for-all".
- Set memberships are dynamic, meaning any changes to the data or relationships will automatically propagate and modify set memberships.
- Any changes to set memberships automatically will apply to all constraints it is a member of. Inferences and retractions will occur automatically.

```
SET case = AutoCase@eg.maximal.co;
SET driver = case.Drivers;
SET mvr = driver.MVR;

// m_viol is a set of all major violations in an MVR.
SET m_viol = mvr.Violations WHERE m_viol.Severity = MAJOR;

// m_viol_yr is a set of all major violations in an MVR in the last year.
SET m_viol_yr = mvr.Violations WHERE m_viol_yr.Severity = MAJOR AND
    DATEDIFF(case.EffectiveDate, m_viol_yr.ViolationDate, YEARS) < 1;
```

"case" refers to all objects of type "AutoCase@eg.maximal.co"

"driver" refers to all Drivers in a case. This is determined by the attribute "AutoCase" of the driver. If that changes, membership will change.

"mvr" is the MVR attached to a driver.

"m_viol" is a conditional set of all violations in an MVR with Severity code of MAJOR. If the Severity is changed to MINOR, the membership will change.

"m_viol_yr" is a conditional set wehre Severity is MAJOR and the violation date is within the last year from effective date.

Decision Logic

maximal

Constraints

- All the eligibility criteria are modeled as constraints as shown below.
- The model assumes a case is eligible by default, unless some criteria is not met.
- Dynamic sets and operations on them are first-level constructs in the model.

```
CONSTRAINT Calc C1 STRICT "Age calculation"
  driver.AgeAtEffective = DATEDIFF(case.EffectiveDate, driver.DOB, YEARS);

CONSTRAINT Eligibility C0 DEFAULT "A case is eligble by default."
    case.IsEligible;

CONSTRAINT Eligibility C1 "A case is eligible only if all drivers are eligible."
    case.IsEligible => driver.IsEligible;

CONSTRAINT Eligibility C2 "Driver must be 18 or over to be eligible."
    driver.IsEligible => (driver.AgeAtEffective >= 18);

CONSTRAINT Eligibility C3 "Driver must have 5 or less major violations."
    driver.IsEligible => SSIZE(m_viol) <= 5;

CONSTRAINT Eligibility C4 "Driver must have 2 or less major violations in the last year."
    driver.IsEligible => SSIZE(m_viol_yr) <= 2;

CONSTRAINT Eligiblity C5 "Driver must have an active license."
    driver.IsEligible => NOT mvr.DLSuspended;
```

Computes age from effective date and date of birth using DATEDIFF function.

Sets case.IsEligible to true, by default. If any other constraints set it to false and a contradiction occurs, this default is withdrawn.

If any driver.IsEligible becomes false, case.IsEligible also becomes false.

If driver is less than 18, driver.IsEligible becomes false, which will also make case.IsEligible to be false.

C3 and C4 are constraints on number major violations, total and within last year. They use the Set Size (SSIZE) function. As the set memberships change, that changes dynamically.

Maximal API: Create a New Case


```
[
  {
    "ObjectType": "AutoCase@eg.maximal.co",
    "CaseID": "AC00001",
    "EffectiveDate": "2021-03-27",
    "Drivers": [
      {
        "FirstName": "John",
        "LastName": "Test",
        "DLNumber": "B534521CA",
        "DOB": "2000-01-01",
        "MVR": {
          "ReportDate": "2021-03-27",
          "DLSuspended": false,
          "Violations": [
            {
              "ViolationDate": "2020-04-20",
              "ViolationCode": "V0031X",
              "Severity": "MAJOR"
            },
            {
              "ViolationDate": "2021-01-09",
              "ViolationCode": "V0042C",
              "Severity": "MAJOR"
            },
            {
              "ViolationDate": "2020-08-16",
              "ViolationCode": "V0031X",
              "Severity": "MAJOR"
            }
          ]
        }
      }
    ]
  }
]
```


POST: *{{endpoint}}/scope/new?proj=AutoInsurance*

- A new ”scope” is created with an initial set of objects instantiated.
- A “scope” maps to one constraint network underneath. All related objects are added into one scope for evaluation. In this model, a scope is essentially all objects related to one auto insurance case.
- Objects are sent in as a JSON body.
- Returns the scope data with its Id that is used for follow up calls.

- Based on the data, case eligibility is derived to be false.
- The constraint “Eligibility.C4” says that for a driver to be eligible, the driver must not have more than 2 major violations. Here there are 3. Hence driver.IsEligible is set to false, which in turn makes case.IsEligible to be false.

```
CONSTRAINT Eligibility C4 "Driver must have 2 or less major violations in the last year."
    driver.IsEligible => SSIZE(m_viol_yr) <= 2;
```

Maximal API: View All Objects

GET: *{{endpoint}}/scope/objects?proj=AutoInsurance&scope=10000001&includeRels=true*

- Returns objects from the requested scope Id.
- *includeRels* specifies whether all relations should be returned.
- All system generated ObjectIDs are also returned with each object.
- External systems can also specify external Ids for objects when they are created.
- Either of these ids can be used to refer to any objects in the system.


```
[
  {
    "ObjectType": "AutoCase@eg.maximal.co",
    "Drivers": [
      {
        "ObjectType": "Driver@eg.maximal.co",
        "ObjectID": 10000001,
        "ObjectTypeID": 10000002,
        "FirstName": "John",
        "DLNumber": "B534521CA",
        "DOB": "Sat Jan 01 00:00:00 UTC 2000",
        "MVR": {
          "ObjectType": "MVR@eg.maximal.co",
          "ReportDate": "Sat Mar 27 00:00:00 UTC 2021",
          "ObjectID": 10000001,
          "Violations": [
            {
              "ViolationDate": "Mon Apr 20 00:00:00 UTC 2020",
              "ObjectType": "Violation@eg.maximal.co",
              "ViolationCode": "V0031X",
              "ObjectID": 10000001,
              "ObjectTypeID": 10000004,
              "MVR": 10000001,
              "Severity": "MAJOR"
            },
            {
              "ViolationDate": "Sat Jan 09 00:00:00 UTC 2021",
              "ObjectType": "Violation@eg.maximal.co",
              "ViolationCode": "V0042C",
              "ObjectID": 10000002,
              "ObjectTypeID": 10000004,
              "MVR": 10000001,
              "Severity": "MAJOR"
            },
            {
              "ViolationDate": "Sun Aug 16 00:00:00 UTC 2020",
              "ObjectType": "Violation@eg.maximal.co",
              "ViolationCode": "V0031X",
              "ObjectID": 10000003,
              "ObjectTypeID": 10000004,
              "MVR": 10000001,
              "Severity": "MAJOR"
            }
          ],
          "ObjectTypeID": 10000003,
          "DLSuspended": false
        },
        "LastName": "Test",
        "AgeAtEffective": 21,
        "IsEligible": false,
        "AutoCase": 10000001
      }
    ],
    "ObjectID": 10000001,
    "ObjectTypeID": 10000001,
    "CaseID": "AC00001",
    "IsEligible": false,
    "EffectiveDate": "Sat Mar 27 00:00:00 UTC 2021"
  }
]
```

Maximal API: Ask for Explanations

GET: *{{endpoint}}/scope/explainVariable?proj=AutoInsurance&scope=10000001&variable=AutoCase@eg.maximal.co:10000001:IsEligible*

- *variable:* object-type:object-id:attribute format
- Returns all constraints and terminal input variables that explain the inference for the decision variable.
- In this case, the constraint Eligibility.C4 and three major violations in the MVR report provide explanation for why IsEligible is false.
- If there are multiple (redundant) explanations, this method will return them all.
- There is another method to fetch the entire explanation path, not just the terminal variables.


```
[
  [
    {
      "variable": "Violation@eg.maximal.co:10000003:Severity",
      "value": "MAJOR",
      "asserted": true
    },
    {
      "variable": "Violation@eg.maximal.co:10000002:Severity",
      "value": "MAJOR",
      "asserted": true
    },
    {
      "variable": "Violation@eg.maximal.co:10000002:ViolationDate",
      "value": "2021-01-09T00:00:00.0Z",
      "asserted": true
    },
    {
      "description": "A case is eligible only if all drivers are eligible.",
      "active": true,
      "constraint": "Eligibility.C1",
      "priority": "NORMAL"
    },
    {
      "description": "Driver must have 2 or less major violations in the last year.",
      "active": true,
      "constraint": "Eligibility.C4",
      "priority": "NORMAL"
    },
    {
      "variable": "Violation@eg.maximal.co:10000003:ViolationDate",
      "value": "2020-08-16T00:00:00.0Z",
      "asserted": true
    },
    {
      "variable": "AutoCase@eg.maximal.co:10000001:EffectiveDate",
      "value": "2021-03-27T00:00:00.0Z",
      "asserted": true
    },
    {
      "variable": "Violation@eg.maximal.co:10000001:Severity",
      "value": "MAJOR",
      "asserted": true
    },
    {
      "variable": "Violation@eg.maximal.co:10000001:ViolationDate",
      "value": "2020-04-20T00:00:00.0Z",
      "asserted": true
    }
  ]
]
```

Maximal API: Add or Modify Objects

```
[
    {
        "ObjectType": "Violation@eg.maximal.co",
        "ObjectID": 10000001,
        "Severity": "MINOR"
    }
]
```

POST: {{endpoint}}/scope/updateObjects?proj=AutoInsurance&scope=10000001

- Here we update one of the violations to Minor from Major.
- This change gets propagated through constraints resulting in inferring case eligibility to be true. That's because the driver no more has more than 2 major incidents in a year.

Summary

- A decision model is Maximal has two parts:
  - Data Model
  - Logic Model or Constraint Model.
- Once these two are defined, you have a decision service up and running. You can access it via an API.
- All data you create, modify or infer via constraints is persisted in a relational database.
  - You can run queries against those tables for reporting or building applications.