XMLdation validation service has been able to support every type of rule so far, as long as the logic of the rule is clear.

The validation can be divided into three parts

- XML well-formedness validation
- Schema validation
- Business rule validation

Schema validation

Schema validation is simply done by comparing the content in inputted XML against the schema of selected validation pipe. Because schema itself includes a set of restrictions this is fairly easy to do. For example following is defined in ISO's pain.001.001.03 schema:

```
...
<xs:complexType name="PaymentInstructionInformation3">
 <xs:sequence>
   ...
   <xs:element name="NbOfTxs" type="Max15NumericText"/>
   ...
 </xs:sequence>
</xs:complexType>
...
<xs:simpleType name="Max15NumericText">
 <xs:restriction base="xs:string">
   <xs:pattern value="[0-9]{1,15}"/>
 </xs:restriction>
</xs:simpleType>
...
```

(Myriad amount of other content is omitted in above example)

This means that that complexType *PaymentInstructionInformation3* defines an element named *NbOfTxs*, and its type is set as *Max15NumericText*. This simpleType is defined later in the schema, and following restriction is set for it: "<xs:pattern value="[0-9]{1,15}"/>", this regex means that all elements with that simpleType can only consist of numbers ([0-9]) and have minimum of 1 and maximum of 15 characters in it ({1, 15}). XMldation service uses these instructions to know whether that element's usage is valid.

Schema consists of plethora of these kinds of restrictions, more information about schemas generally can be found from W3's schema section.

Schema gives the XML file a proper structure and defines a basic set of restrictions for element. However, schema is very limited in the style of rules it can validate. In addition to defining the basic structure for XML file, with schema it is possible to restrict following for an element:

- Length
- Occurrence amount

- Allowed characters (with regex)
- Possible values
- Type of an element (e.g. string, decimal, dateTime)

Please note that there are services offering schema validation, for example a popular program XMLspy. The strength of XMLdation schema validation is that we are able to give multiple schema errors at once and show them in a clear manner.

Business rules validation

As described above, schema can only go so far when it comes to validating elements. Below are example of styles of rules which are not possible to be validated by schema:

- If-Else statements. When certain data is given in one element, another data has to exist
    - ? Using certain element prohibits the usage of another element elsewhere in file
    - ? When certain data is given in one element, another type of data is mandatory/prohibited
    - ? Using certain attribute restricts the possible data of the element
- Arithmetic sum of data in elements
- Check digit matches certain algorithm
- Rules which do not return error, but a notice/info instead. In some cases it is beneficial to inform a user when certain data is used

XMLdation validation service uses OCL for business rules validation, and with it we are able to make validation rules for every desired situation. Other advantages of using only OCL to create business rule validation rules is:

- The text in error / notice messages can be written by us with the collaboration client's specialist
- Original schema in pipe can be left untouched, allowing us to document rules well
- Allowing us to modify the rules in pipe easily and swiftly.

Combining schema and business rules validation

When a file is inserted into validation pipe only schema validation is initiated at first. This is to ensure that given file is in fact in XML format and has the correct structure for the schema. Schema has to be used as a basis for business rule (OCL) validation, so we are able to give meaningful results only after when the structure of the file is correct. When a file / part of the file has a correct structure based on it's schema, business rule validation is executed.

Possible schema errors and all possible business rule errors are then shown in a single validation report.