

# Parties

This presentation has 2 purposes:

1. to present a brief overview of the development tools.
2. to present the first standard module of our finance application: the parties.

Your first view is on the Eclipse development platform where Topbraid, the modelling tool is embedded. The developer works mainly in this environment.

Here data is modelled, properties are allocated to concepts, restrictions and inferences of individuals are set-up, queries are tested and drawn, forms get their shape, functions are written, automated and advanced procedures are designed, instances are created, problems and errors are consulted.

Also the stored data is verified. The storage occurs in the most generic possible way. Meaning 3 columns: subject, predicate and object.

Physically this information is stored in an RDF-, xml-, owl- file or a database.

From the FAQ page you can find more information on the storage.

The user never sees this modelling environment.

Although the developer does not use programming techniques or tools, the environment might frighten at first sight.

The data model is shown straight into Ensemble.

The part which is seen by the user of the application is called Ensemble.

The functions available to the user may be huge or basic depending on the wishes of our clients.

Example: in our sample Parties application, the coordinates of the residence addresses are retrieved, the addresses are shown on a map. This might be performed in batch to gain resources. Also, only commercial people in front-office are interested in this information.

Ensemble is the user interface for the clients via the internet, the front-offices and the back offices via the intranet connections.

The available options and information shall of course be different according to the user.

The client will only be able to see his own information.

The branches will have access to, depending on the sales policy, the own customers, or the customers of a whole brand.

The back-office people do in principle not need to see the addresses of customers.

All components are moveable and sizeable.

In the tree component we browse and select the concepts which are necessary to manage the parties involved in financial transactions and securities handling. Those concepts are called classes in ontology vocabulary.

Once a selection is made in the tree component, the *results grid* displays a predefined number of members/elements of that class. In ontology vocabulary we call these members individuals.

The search component is also adapted according to the properties of the selected concept.

When an specific individual or a specified group of individuals has to be filtered in the *results grid*, criteria on the properties are set in this component.

Once an individual is selected in the *results grid*, its properties are displayed in the *form*.

The ontology Parties is one part of the federation of ontologies which make up the finance ontology.

The concept of party, represented by an ontology and a class comprises notably:

- servicing company: the legal entity using the ontologies and rendering the financial services
- principals: the ones for who's account an in who's name the administration is kept: normally a financial institution
- clients: the customers of the principals
- custodians: the ones who hold the securities of the clients in custody
- paying agents, the ones who pay the dividends for account of the issuers, ...
- ...

The development of such an ontology is never finalized. The process is a iteration of reviews with enhancements and deeper detailing at each review. At this moment the level of detail takes into consideration the known international norms and the ISO 20022 message definitions.

Taking into consideration these message definitions means we are able to read the messages straight or with only few modifications into the ontology.

The definitions of the classes, thus the concepts with their properties, will be treated under the chapter of the documentation.

We start by reviewing the classes as they present themselves in the application, thus alphabetically.

## **The blacklists.**

The external lists contain the identity of parties or countries which are internationally signalled terrorists or

money launderers: the US- Office of Foreign Asset Control (OFAC) maintains this list , aviation companies with embargo (EU) or countries on which there is an embargo (UN).

The OFAC list is a real-time updated list.

The internal list contains the identity of known problem parties following the criteria established by our principals.

Batch procedures can run against existing customers on a regular basis to compare them with updated blacklists.

## **The citizenship information**

Contains the information which allows to determine the juridical capacity of a party.

The nationality is necessary to determine the legislation which is applicable on the relations.

Also, as you can see from the link to the country, there is a possibility to derive the flag minor from the country information.

Other restrictions than the minor one may be applicable, we shall create a new instance for each person having a restriction different from the combination nationality/minor ones.

We enter on that personalized citizenship individual a link to the documentation on the restriction. This documentation is mostly a Court judgement which cannot be standardized.

<http://fadyart.com/files/VanderlindenJoris.txt>

## **Compliance questionnaire**

The questionnaire gives an indication whether the client can be accepted right away to open a relationship or whether an “enhanced due diligence” investigation is necessary by the compliance department (Anti-Money Laundering unit) indicated in the procedures.

These departments can get automated lists or e-mails with the responses to these questionnaires, as well as from the clients on blacklists.

If one indication is true, the client does not get e.g. an account number straight away, we will tell the client he will get a response later on.

## **Compliance validation profile**

If the questionnaire does not reveal suspicious actions, the compliance profile can be filled based on the filed documentation and the procedures on the client acceptance policy.

Each client relation has to be documented according to the “Know Your Customer” rules to prove the recorded identity.

A link to that scanned documentation is registered here.

Comments can be filled as with each individual member of a class.

## **Contact information**

Postal addresses are kept in a separate class.

Here are additional addresses for specific purposes, specific frequencies or specific portfolios or accounts.

## **Country and residential status**

This information is crucial for income tax calculations.

Income tax calculations are executed when determining the foreign and local withholding tax from revenues. Example: a US dividend will get different US withholding rates according to the country of residence of the beneficial owner.

Further, concerning withholding taxes, the country of fiscal residence of the principal (the bank where the client has the customer role) will be compared with the country of residence of the client to determine whether the withholding tax is local or foreign.

If no entry is found for any party, the country of the residential address will determine the country of fiscal residence.

## **Customer acceptance processes**

Those processes and procedures are adaptable to the specific legislation of the principals and the principals policies.

The samples are interesting because they illustrate warning issues. Indeed, as the organizational ontology is not linked yet, the object of the triples, or the right side of a relationship if you wish at this time, cannot be completed as defined in the class.

## **Generic identification**

The purpose is to identify the beneficial owner(s) of an account.

The class is inspired by the ISO 20022 messages.

## Identification

The purpose is to identify the communication partners in financial transactions.

This class is inspired by the ISO 20022 messages. Indeed, with different communication partners, different identifiers are necessary.

This one is normally used to specify the 3 normally used in automated transactions between financial institutions:

1. Id by BIC or BEI code, also called SWIFT-address
2. Id by reference to a proprietary scheme: example the Dun & Bradstreet codifications  
This method of identification refers to class "Generic identification" where the scheme and identifier are registered.
3. Id by name and address: here reference is made to the "Name and address" class where an individual will have to be identified.

## MiFID questionnaire

The questionnaire permits the proposition of an investment profile.

To that profile a model portfolio is linked.

A customer can link this profile or another profile to any portfolio he will open.

All questions have to be filled if a proposition is expected.

## Name and address

Each party needs at least 1 registration address.

This address can also be used for the normal correspondence of free information.

The mailing indicator is then also set.

For mailing, a specific denomination can be used. Example:

Soccer club Anderlecht (Specific denomination)  
P/A Mr Albert Vandestock

Type of address is "Deliver" e.g. in case of physical deliveries of securities to a custodian.

When the address is complete, the longitude and latitude figures are filled and the address can be viewed on the map.

## Proprietary identification

See Identification

## Party: individual person and organization.

What is seen above has to be related to a party.

That's what the classes here are meant for.

You can try to link existing information in classes treated above to a Party, start from scratch or make a mixture.

Here we go with an example individual person: see separate instruction video.

## **Value partition.**

This big superclass contains the values and explanations which are referred to from other classes. Of course these classes are also documented.

## **Graph editor and query**

With each module, I intend to demonstrate the building of a query.

Of course, you have plenty of reports which can be drawn but a query on the fly of intelligent information should be an automation whenever one feels the need to.

Today the example concerns an ordered list of the procedure steps on the customer acceptance steps for my business line: retail banking.

We select tab Graph Editor and Query

The trick is to show the machine which information you want in the result.

Thereto we drag an individual from the process class to the Graph pane.

We are interested in the outgoing properties: Comment, since that property contains the explanation on the step, the label since it characterizes the step, the sequence since we want to know what step comes after which other, who has to perform it because I'm interested only in my own work.

So, let's select these 3 from the example process.

One can freeze the lay-out, zoom in and out.

Since we are not interested in a particular procedure step, neither in a particular sequence, neither in a particular comment, we generalize these properties. Pay attention not to generalize the “performed by” property since we only want our own retail business line's steps.

We don't find the variable names, generated when generalizing the properties very relevant, so we change to “sequence” and “detailed explanation” the variables bound to respectively the properties “Procedure sequence” and “Comment”.

We run the query by selecting the appropriate button.

The result is shown in the result grid.

We want the results sequentially ordered, so we click the header above the sequences.

The sequence has to be mentioned in the first column, so we move it there.

The width of the columns is adapted by dragging the borderlines of the titles.

The result suits us and we would like to copy sequence points 1 to 12 into a spreadsheet.

So we select those related procedure steps and click the copy button.

Paste into your spreadsheet.

Since we expect we'll need the query in the future, we save it by pressing the save button and allocating a name.

## **The documentation**

The documentation is the last topic on this module.

It is produced as ontology documentation on the website :

Hope this was useful and that we may assist you in future developments with these tools of the future.