Navigator Concept

Table of Contents

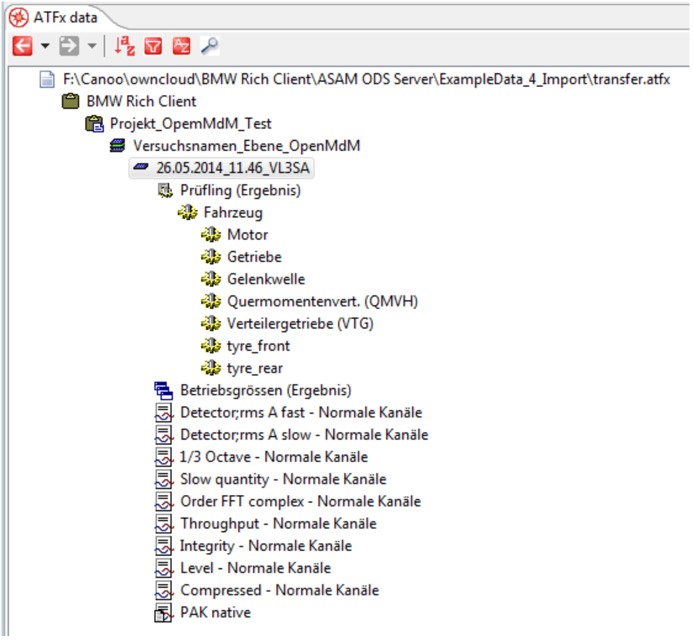
This document explains one central use case of the OpenMDM application: Exploring and finding measurement data of commissioned tests. At first the implicated requirements concerning navigation and search are described. Afterwards a new solution is proposed and explained in detail. Note that the analyzed OpenMDM client is a test release of 13th May 2014.

Chapter 1. Navigational requirements

The amount of data which has to be dealt with is immense and constantly growing. Each day new projects are set up, tests are commissioned and measurement results are reported. Therefore the application has to provide filter capabilities which allow to advance quickly to the data of interest. Furthermore switching between related tests and measurements (e.g. all tests on the same unit) should be possible. Easily retrieving previously visited projects, tests and measurement results in the data space would be appreciated.

1.1. Realization in OpenMDM 4

The central component within the current OpenMDM application for browsing the test projects, tests and measurement results and all associated meta data is the *navigator* which organizes all this data in a tree structure. This approach is suitable as many one-to-many relations exist in the test domain: One project contains multiple tests, each test consists of several test steps which lead to various measurement results.



- F:\Canoo\owncloud\BMW Rich Client\ASAM ODS Server\ExampleData_4_Import\transfer.atfx
  - BMW Rich Client
    - Projekt_OpemMdM_Test
      - Versuchsnamen_Ebene_OpenMdM
        - 26.05.2014_11.46_VL3SA
          - Prüfling (Ergebnis)
            - Fahrzeug
              - Motor
              - Getriebe
              - Gelenkwelle
              - Quermomentenvert. (QMVH)
              - Verteilergetriebe (VTG)
              - tyre_front
              - tyre_rear
          - Betriebsgrössen (Ergebnis)
          - Detector;rms A fast - Normale Kanäle
          - Detector;rms A slow - Normale Kanäle
          - 1/3 Octave - Normale Kanäle
          - Slow quantity - Normale Kanäle
          - Order FFT complex - Normale Kanäle
          - Throughput - Normale Kanäle
          - Integrity - Normale Kanäle
          - Level - Normale Kanäle
          - Compressed - Normale Kanäle
          - PAK native

Figure 1. Navigator in OpenMDM 4

As with the constant growing of the data the tree becomes more and more complex and confusing. To master the amount of data some features have been combined in a navigator toolbar:

- History

  Up to five lately visited nodes are held in a history and can be chosen via a dropdown menu. The corresponding node is immediately selected. Furthermore a basket is provided in which the user can actively put nodes he needs to have at hand.

- Alphabetic sort across all nodes

  A toggle button allows to switch between sorting all nodes according to creation date and or name respectively. A certain node can easier be found if all nodes on each level are sorted alphabetically, assuming the name of the node is known.

- Filter

  A filter dialog can be opened which allows to specify which naming conventions nodes need to comply with in order to be visible in the tree structure. For each node type different name pattern can be defined.

- Sort according node type

  Nodes of the same type can be grouped. This can be helpful, when the node type is known and also facilitates the manual search.

- Global search

  A search dialog enables the user to find nodes whose name contains a given character string. It is even possible navigate to the next matching hit.

1.2. Strengths and weaknesses

The history feature is very powerful. The user can move to another node, view or change some details and be certain that he can quickly return to the previous node. This is possible out-of-the-box without needing to tag or place the node into the shopping basket.

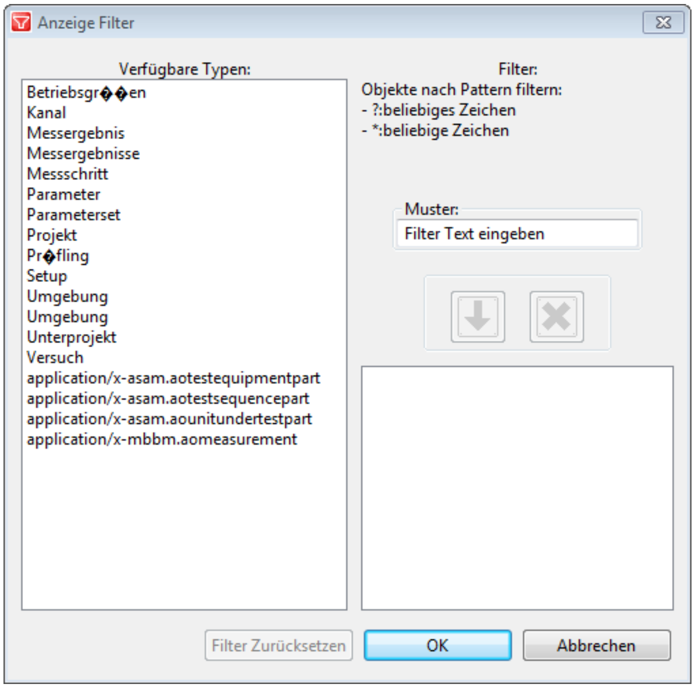
In large trees the alphabetic and typed sort functionality can leverage the effort to find a special node, given that the name or the type respectively are known.

Sadly the buttons for the two sort actions have quite similar icons and thus are easy to confuse with each other.

The currently applied filters and the fact that filters are active are hidden behind a toolbar button. The user can not be sure without checking whether the navigator displays all potentially available data.

As another consequence quick changes of the filter parameters are impossible.

Furthermore the filter configuration dialog has some weaknesses concerning appearance and user interaction:



Objekte nach Pattern filtern:
- ?:beliebiges Zeichen
- *:beliebige Zeichen

Muster: Filter Text eingeben

Figure 2. Filter configuration dialog for navigator

Too many vertical lines distract the user.

Unnecessary titled and untitled borders only add visual clutter to the dialog.

The filter value has to be entered in a textfield labeled 'Sample' (german: 'Muster'). The user would expect this to be an example expressing the syntax to be used. But then a non-editable label should be used for this informational text.

Before entering search patterns into the textfield it has to be cleared because the hint text is not realized as a placeholder but written into the textfield. This has to be done every time the dialog is opened which is quite tedious.

When entering filter patterns then wildcards supported but their usage is enforced. If only a substring of the node name is known, this string has to be enclosed with special characters (*) to allow multiple random characters before and after and thus produce a reasonable result.

No summary of the currently applied filters is presented. In order to find out all filters which are actually applied all types have to be checked.

Adapting a filter pattern for a given type means removing and typing it again which is quite annoying for the user.

In contrast to this clearing all filters at once is supported by an extra button in the configuration dialog.

The global search is the fastest way to get to the node of interest.

Even though it is a commonly used feature it is hidden behind a toolbar button.

The usage of wildcards is again enforced if only a substring of the node name can be entered.

The user can specify if he wants to apply a breadth-first search algorithm. But the dialog does not express what not checking this option means.

In order to fasten the search the corresponding dialog allows to restrict the search to the sub-tree below the currently selected node.

The number of nodes matching a certain search pattern is not indicated. The user has to click through all hits.

The dialog allows to jump to the next hit. But no jumping back is supported. Furthermore when all matching nodes have been visited the search cannot be restarted from the top. Instead the search dialog needs to be closed, reopened and the search pattern entered again.

When a node in the navigator tree is expanded then for all child nodes, even for leafs a plus icon is displayed making the user believe that also those nodes can be expanded. It is quite annoying for the user to find out which nodes are really expandable.

Chapter 2. Search requirements

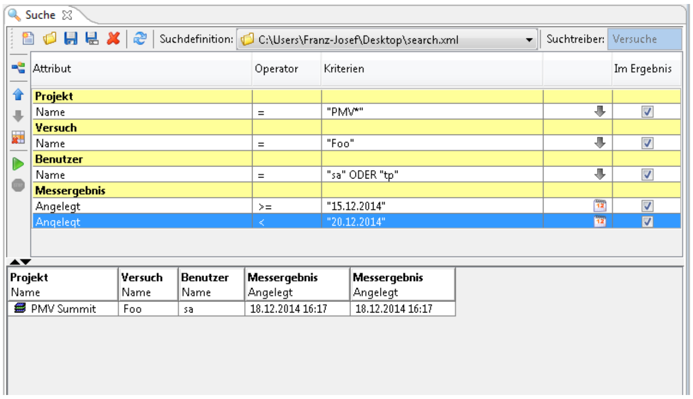
In order to have random access to the measurement data a powerful search component is needed which has to allow specifying all known information as search criteria that is somehow related to the data sets of interest. For example it should be possible to name the project or the test status and then find all measurement results which are connected to matching project and test entities.

Furthermore the search module has to allow to refer different data locations (ASAM ODS systems) in order to include or exclude archives.

As some searches will be executed regularly (e.g. to check if all today’s tests passed successfully) the application has to support loading and saving of search configurations (search attributes, their order and concrete search values). Thereby predefined search configurations shall be available, but the user may also define his private ones. The user may then publish them to a specific user or a whole user group.

2.1. Realization in OpenMDM 4

A highly sophisticated search module has been created for the OpenMDM 4 application.



Suchdefinition: C:\Users\Franz-Josef\Desktop\search.xml

Suchtreiber: Versuche

Attribut	Operator	Kriterien		Im Ergebnis
Projekt				
Name	=	"PMV*"		☑
Versuch				
Name	=	"Foo"		☑
Benutzer				
Name	=	"sa" ODER "tp"		☑
Messergebnis				
Angelegt	>=	"15.12.2014"		☑
Angelegt	<	"20.12.2014"		☑

Projekt Name	**Versuch** Name	**Benutzer** Name	**Messergebnis** Angelegt	**Messergebnis** Angelegt
PMV Summit	Foo	sa	18.12.2014 16:17	18.12.2014 16:17

Figure 3. A sample search configuration

It allows to load predefined search configurations, thereby distinguishing between system wide and private ones and those imported from the file system. Search configurations can be saved and saved with another name and deleted if no longer used.

In order to start a search the user has to specify which type of data sets he is interested in. This restricts the amount of available search categories (e.g. Project) and search attributes (e.g. Name) therein. For each attribute a pattern can be defined which must be matched or may not be matched, depending on the given operator. A dialog can be opened which lists all available values which have been found in the database for this attribute, which helps the user to enter meaningful values. The search result is presented in a table whose columns are generated from the defined search attributes. But the user can also manually exclude a search attribute.

2.2. Strengths and weaknesses

Allows to load predefined search definitions. Thereby the user can differentiate between system-wide available searches, his own saved searches and search definition files available from the file system.

Current search specifications can be saved and saved with another name.

Delete current search (file) function can easily be confused with remove search attributes or clear search values function.

A confirmation dialog prevents the user from accidentally deleting saved search files.

User can clear search mask entries via context menu action.

User has to actively start search via toolbar button, context menu or Enter key.

After deleting all search attributes a reload of the saved search attributes is not possible. The search module needs to be closed and reopened.

Execution time and number of hits are indicated.

User can stop a long-running search.

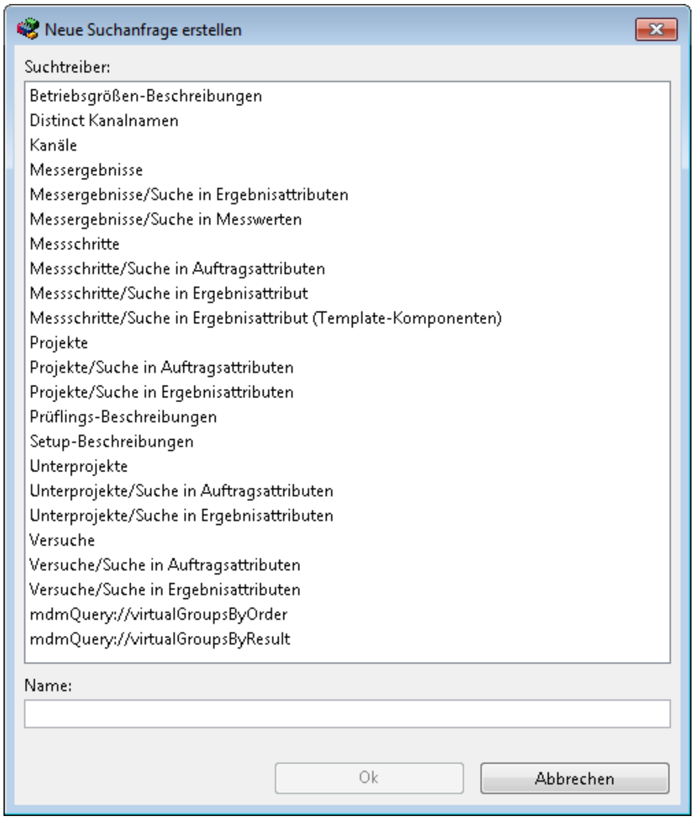
Attributes can be ordered manually in search definition but not in result table.

Result table does not show the whole entity (e.g. test) but only those attributes (as columns) which were defined in the search mask. To get more information either the attributes of the searched entity need to be added as search attributes or the item of the result table needs to be added to the 'Shopping basket' and selected in the navigator tree. Then finally the item can be investigated in the 'Details' view which might also need to be opened first.

All possible values of a certain search argument are listed in a dialog, available via click on the cell in the untitled column.

Search attributes can be manually excluded from result table, so that no column is created for them.

Multiple similar named search drivers confuse the user. He does not know which one to choose.



Neue Suchanfrage erstellen

Suchtreiber:

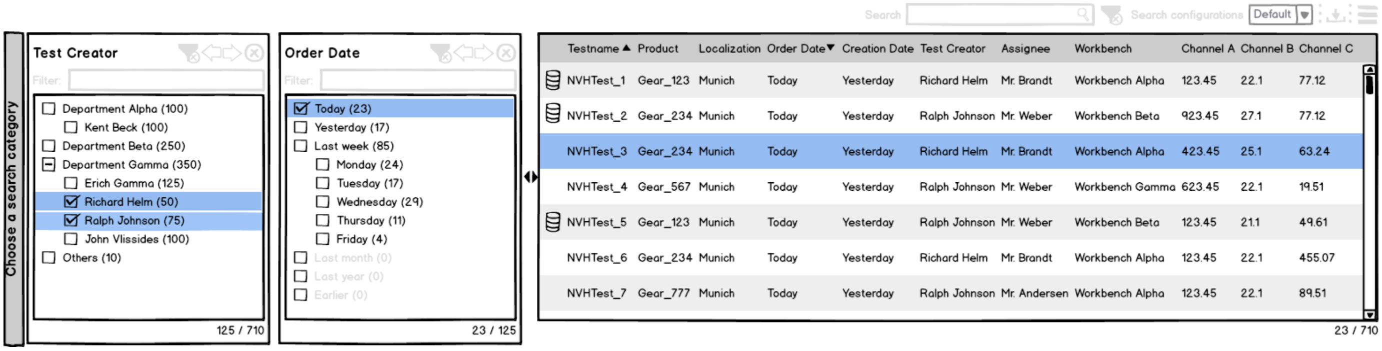
- Betriebsgrößen-Beschreibungen
- Distinct Kanalnamen
- Kanäle
- Messergebnisse
- Messergebnisse/Suche in Ergebnisattributen
- Messergebnisse/Suche in Messwerten
- Messschritte
- Messschritte/Suche in Auftragsattributen
- Messschritte/Suche in Ergebnisattribut
- Messschritte/Suche in Ergebnisattribut (Template-Komponenten)
- Projekte
- Projekte/Suche in Auftragsattributen
- Projekte/Suche in Ergebnisattributen
- Prüflings-Beschreibungen
- Setup-Beschreibungen
- Unterprojekte
- Unterprojekte/Suche in Auftragsattributen
- Unterprojekte/Suche in Ergebnisattributen
- Versuche
- Versuche/Suche in Auftragsattributen
- Versuche/Suche in Ergebnisattributen
- mdmQuery://virtualGroupsByOrder
- mdmQuery://virtualGroupsByResult

Name:

Figure 4. New search request dialog

Chapter 3. Proposed solution: Faceted Search

The following chapter explains the suggested solution which addresses the above mentioned requirements as well as offering the user even more capabilities to explore the data.



Test Creator

Filter:

☐ Department Alpha (100)
  ☐ Kent Beck (100)
☐ Department Beta (250)
⊟ Department Gamma (350)
  ☐ Erich Gamma (125)
  ☑ Richard Helm (50)
  ☑ Ralph Johnson (75)
  ☐ John Vlissides (100)
☐ Others (10)

125 / 710

Order Date

Filter:

☑ Today (23)
☐ Yesterday (17)
☐ Last week (85)
  ☐ Monday (24)
  ☐ Tuesday (17)
  ☐ Wednesday (29)
  ☐ Thursday (11)
  ☐ Friday (4)
☐ Last month (0)
☐ Last year (0)
☐ Earlier (0)

23 / 125

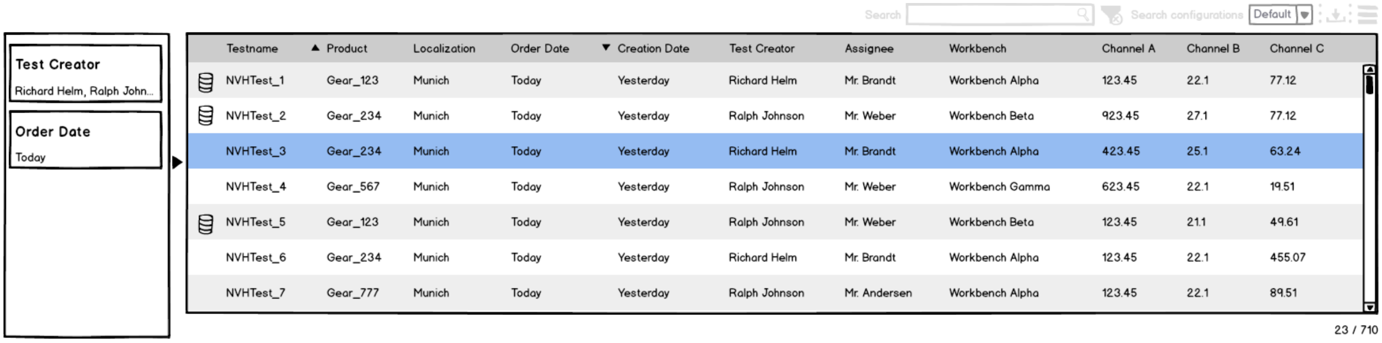
Testname ▲	Product	Localization	Order Date ▼	Creation Date	Test Creator	Assignee	Workbench	Channel A	Channel B	Channel C
NVHTest_1	Gear_123	Munich	Today	Yesterday	Richard Helm	Mr. Brandt	Workbench Alpha	123.45	22.1	77.12
NVHTest_2	Gear_234	Munich	Today	Yesterday	Ralph Johnson	Mr. Weber	Workbench Beta	923.45	27.1	77.12
NVHTest_3	Gear_234	Munich	Today	Yesterday	Richard Helm	Mr. Brandt	Workbench Alpha	423.45	25.1	63.24
NVHTest_4	Gear_567	Munich	Today	Yesterday	Ralph Johnson	Mr. Weber	Workbench Gamma	623.45	22.1	19.51
NVHTest_5	Gear_123	Munich	Today	Yesterday	Ralph Johnson	Mr. Weber	Workbench Beta	123.45	21.1	49.61
NVHTest_6	Gear_234	Munich	Today	Yesterday	Richard Helm	Mr. Brandt	Workbench Alpha	123.45	22.1	455.07
NVHTest_7	Gear_777	Munich	Today	Yesterday	Ralph Johnson	Mr. Andersen	Workbench Alpha	123.45	22.1	89.51

23 / 710

Figure 5. The facet component on the left and the result table on the right

The facet component adds a highly sophisticated filter mechanism to the data table. The facet component contains the *Facet Chooser* and all opened facets - concepts which are described in detail in the next sections.

Because the facet component and the result table are placed in a resizable split pane, the user can freely divide the available screen space between the two UI elements. Thus the user can concentrate either on specifying the filter conditions or browse through the table or operate both.



Test Creator

Richard Helm, Ralph John...

Order Date

Today

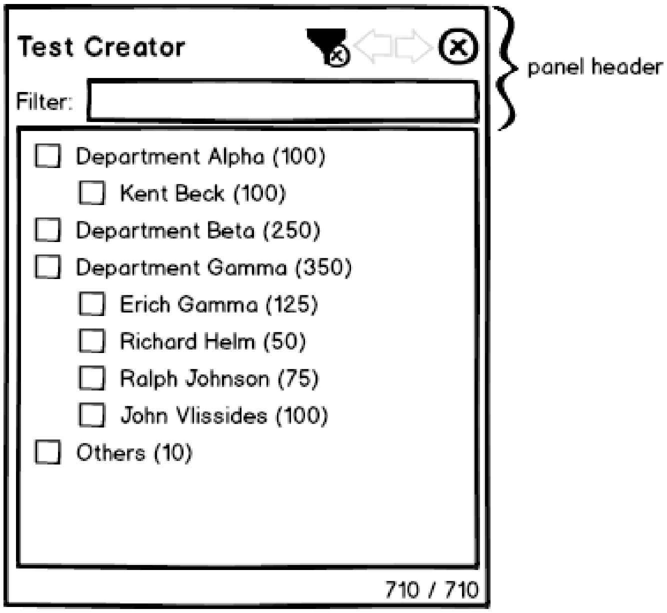
Testname	▲ Product	Localization	Order Date	▼ Creation Date	Test Creator	Assignee	Workbench	Channel A	Channel B	Channel C
NVHTest_1	Gear_123	Munich	Today	Yesterday	Richard Helm	Mr. Brandt	Workbench Alpha	123.45	22.1	77.12
NVHTest_2	Gear_234	Munich	Today	Yesterday	Ralph Johnson	Mr. Weber	Workbench Beta	923.45	27.1	77.12
NVHTest_3	Gear_234	Munich	Today	Yesterday	Richard Helm	Mr. Brandt	Workbench Alpha	423.45	25.1	63.24
NVHTest_4	Gear_567	Munich	Today	Yesterday	Ralph Johnson	Mr. Weber	Workbench Gamma	623.45	22.1	19.51
NVHTest_5	Gear_123	Munich	Today	Yesterday	Ralph Johnson	Mr. Weber	Workbench Beta	123.45	21.1	49.61
NVHTest_6	Gear_234	Munich	Today	Yesterday	Richard Helm	Mr. Brandt	Workbench Alpha	123.45	22.1	455.07
NVHTest_7	Gear_777	Munich	Today	Yesterday	Ralph Johnson	Mr. Andersen	Workbench Alpha	123.45	22.1	89.51

23 / 710

Figure 6. The facet component shows applied filters also in collapsed mode

3.1. Facet

A facet is a filter widget representing one category according to which the data of interest can be filtered. Usually the category correlates with an attribute of the data e.g. zip code when filtering addresses, but might also be an attribute of a related entity e.g. total population of the city. All distinct values of this attribute which can be found in the database are displayed in the facet along with the amount of occurrences.



☐ Department Alpha (100)
  ☐ Kent Beck (100)
☐ Department Beta (250)
☐ Department Gamma (350)
  ☐ Erich Gamma (125)
  ☐ Richard Helm (50)
  ☐ Ralph Johnson (75)
  ☐ John Vlissides (100)
☐ Others (10)

710 / 710

Figure 7. A sample facet

The value in combination with the amount of occurrences is henceforth referred to as a *facet item*. When the number of facet items is too high then they need to be grouped and organized in a tree structure. For example all test engineers can be grouped according to the departments they work in or all production sites in Germany can be assembled under a node of this very name, which, along with other country nodes is placed under a Europe node. Dates and number values need to be gathered under nodes which express date and number ranges respectively. Thereby parent nodes aggregate the data set counts of their children.

NOTE The hierarchical organization gives the user the possibility to quickly advance to the facet items of interest. Furthermore only the child nodes of those nodes need to be loaded which are touched by the user. It is crucial that especially the intermediate nodes bear a commonly understandable, unique label so that the user is not distracted from his goal.

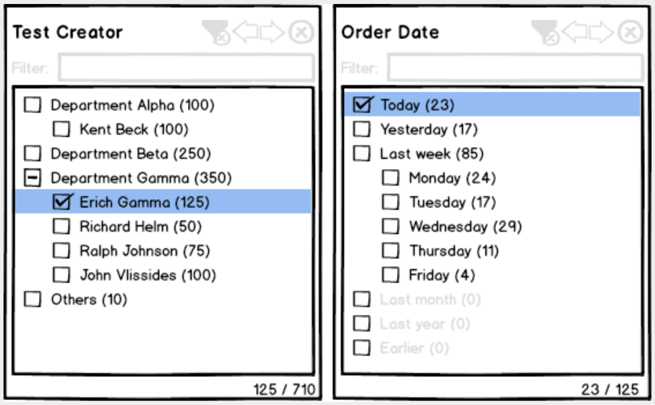
A default facet item (e.g. named Others) has always to be available collecting all the values which cannot be assigned to a specific item e.g. because no value is given. Therefore checking the default option is a good way to find incomplete or outlier data sets.

By selecting a facet item the user can reduce the total amount of data to those data sets which match the selected value in the facet category. Multi selection can be applied by checking two or more facet items. The result set then contains all data sets which match one of the selected values. Selections are propagated to the opened facets on the right invoking a recalculation of the occurrence amounts.

Example

Assume the user wants to find the tests created by a hypothetic test engineer Erich Gamma in the Department Gamma and commissioned today. For this the user opens two facets *Test creator* and *Order date*.

He opens the *Department Gamma* node and selects the *Erich Gamma* facet item which indicates that in total 125 tests have been created by him. Immediately the amounts in the *Order date* facet are recalculated, expressing now the distribution of the 125 data sets onto the temporal facet items. Hence the user immediately knows that Erich Gamma issued 23 tests today. In order to shrink down the result data to these data sets the user just clicks on the *Today* facet item.



Test Creator

Filter:

- ☐ Department Alpha (100)
  - ☐ Kent Beck (100)
- ☐ Department Beta (250)
- ⊟ Department Gamma (350)
  - ☑ Erich Gamma (125)
  - ☐ Richard Helm (50)
  - ☐ Ralph Johnson (75)
  - ☐ John Vlissides (100)
- ☐ Others (10)

125 / 710

Order Date

Filter:

- ☑ Today (23)
- ☐ Yesterday (17)
- ☐ Last week (85)
  - ☐ Monday (24)
  - ☐ Tuesday (17)
  - ☐ Wednesday (29)
  - ☐ Thursday (11)
  - ☐ Friday (4)
- ☐ Last month (0)
- ☐ Last year (0)
- ☐ Earlier (0)

23 / 125

Figure 8. The selection propagation to facets on the right

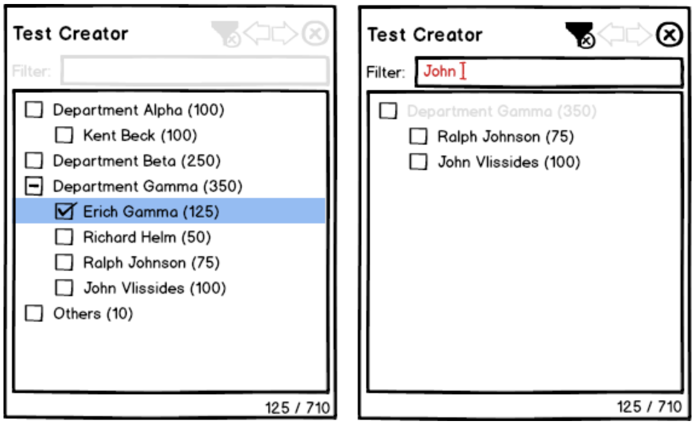
The sum of data sets in the result table before and after applying the filters of the facet is indicated at the bottom.

NOTE If no item in the tree is selected, then no special filter applies from this facet. It has the same meaning as selecting all available items. Furthermore selections of parent nodes causes a selection of all corresponding child nodes.

Facet items with an occurrence count of zero are grayed out as they do not contribute to the data filtering and therefore should not distract the user.

For the same reason the widgets in the facet header are displayed transparent when they are not used. They become opaque as soon and as long as the cursor is hovered above the facet header or one of its widgets gains focus.

One of those widgets is a filter textfield which allows to filter the facet items. Entered text is matched against the facet item names. Thereby wildcards are supported but not enforced i.e. there is no need to add a wildcard character (*), but it does not harm either. In order not to loose the context the whole paths from the matching nodes to the root node have to remain visible. The nodes which only provide context are grayed out.



☐ Department Alpha (100)
  ☐ Kent Beck (100)
☐ Department Beta (250)
⊟ Department Gamma (350)
  ☑ Erich Gamma (125)
  ☐ Richard Helm (50)
  ☐ Ralph Johnson (75)
  ☐ John Vlissides (100)
☐ Others (10)

Filter: John

☐ Department Gamma (350)
  ☐ Ralph Johnson (75)
  ☐ John Vlissides (100)

Figure 9. An unfiltered and filtered facet

NOTE Current selections are kept while filter text is entered.

The facet toolbar features the following buttons:

- One button to clear the filter textfield as well as all selections in the facet
- One button to switch places with the facet on the left. This button is disabled when there is no facet on the left.
- One button to switch places with the facet on the right. This button is disabled when there is no facet on the right.
- One button to close the facet. The facet is removed from view, all selections are cleared and the facet is available again in the facet chooser component.

NOTE The reordering of the facets can also be realized via drag and drop.

3.2. Facet Chooser

The *Facet Chooser* is an animated panel which can slide into view when needed and shrink to a minimum size when the user shifts his attention to other parts of the application.

The *Facet Chooser* lists all facets available for data drill down.

NOTE As a result of user tests facets are translated to *Search Categories*, a term much easier to understand.

For each facet a clickable link is provided stating the facet’s unique name. When the user clicks on it, the link is removed from the *Facet Chooser* and the corresponding facet is opened on the very right of all facets. Facet links may also be displayed in a tree view in order to group them hierarchically. Note that only leaf nodes would be available as facets, root and intermediate nodes would serve only to structure them.

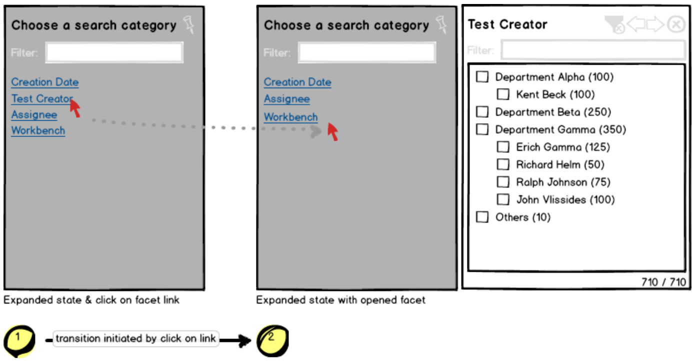


Figure 10. Opening a facet with pinned Facet Chooser

The *Facet Chooser* is placed on the left side of the facet container. To let the panel expand to the right no click is needed - simply hovering with the cursor above the panel is enough. As long as the cursor stays above it, the panel remains in this expanded state. In order to keep the *Facet Chooser* from minimizing again, it can be pinned by clicking the corresponding toggle button in the panel header. Clicking the toggle button again reactivates the default behaviour.

NOTE Unexperienced users may have a hard time to find the panel. Thus for a user who has to work with the facet search approach for the first time, the facet chooser shall be opened and pinned initially. He then can decide for himself, if and when the facet chooser needs to much space and consequently unpin it. As the minimizing of the facet chooser is an animated transformation the user witnesses where the panel remains available for further use.

Additionally the panel header contains a filter textfield allowing to reduce the amount of facet links displayed. Wildcards are supported, but not necessary. The filter is case-insensitive.

All widgets in the panel header are displayed slightly transparent by default. They become opaque as soon and as long as the cursor is hovered above the panel header or one of the widgets gains focus. Thus the widgets remain available but do not distract the user's eye as long as they are not actively used.

3.3. Toolbar

The toolbar above the result table provides quick access to features commonly used when working with tables in the OpenMDM application. These features are convenient even if no facets are used.

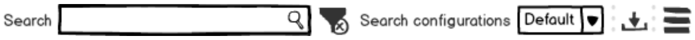


Figure 11. Toolbar of the facet component

A filter textfield allows to enter the name (or parts of it) or the unique id of a data set. This is a first suggestion, presuming that all data sets provide a name attribute. It may also be useful to allow filtering for vehicle ids - this has to be discussed with the application end users. User tests revealed that also date values like Today and Yesterday as well as workbench and production series ids, clerk names or V-numbers are search candidates. In any case the textfield should learn the search terms and suggest them automatically when the user begins to retype them.

When clicking the *ClearAll* button in the toolbar the filter textfields in the toolbar and in the facets are cleared as well as all facet item selections. All currently opened facets remain open but do not apply any filter and therefore the result table lists all data sets.

The *Search Configuration Chooser* is an extended, filterable combobox which allows to load and save facet component and table compositions. This comprises the opened facets, their order, facet item selections for each facet and the toolbar filter textfield entry, as well as the table state (order of the columns, column sort state, visible and hidden columns, column widths). At least one system wide default composition will always be provided. The default compositions will be negotiated with the OpenMDM application end users. Compositions can be switched by simply selecting another item in the combobox. A special menu entry serves for saving the current facet component and table composition. Please refer to the Appendix for more information.

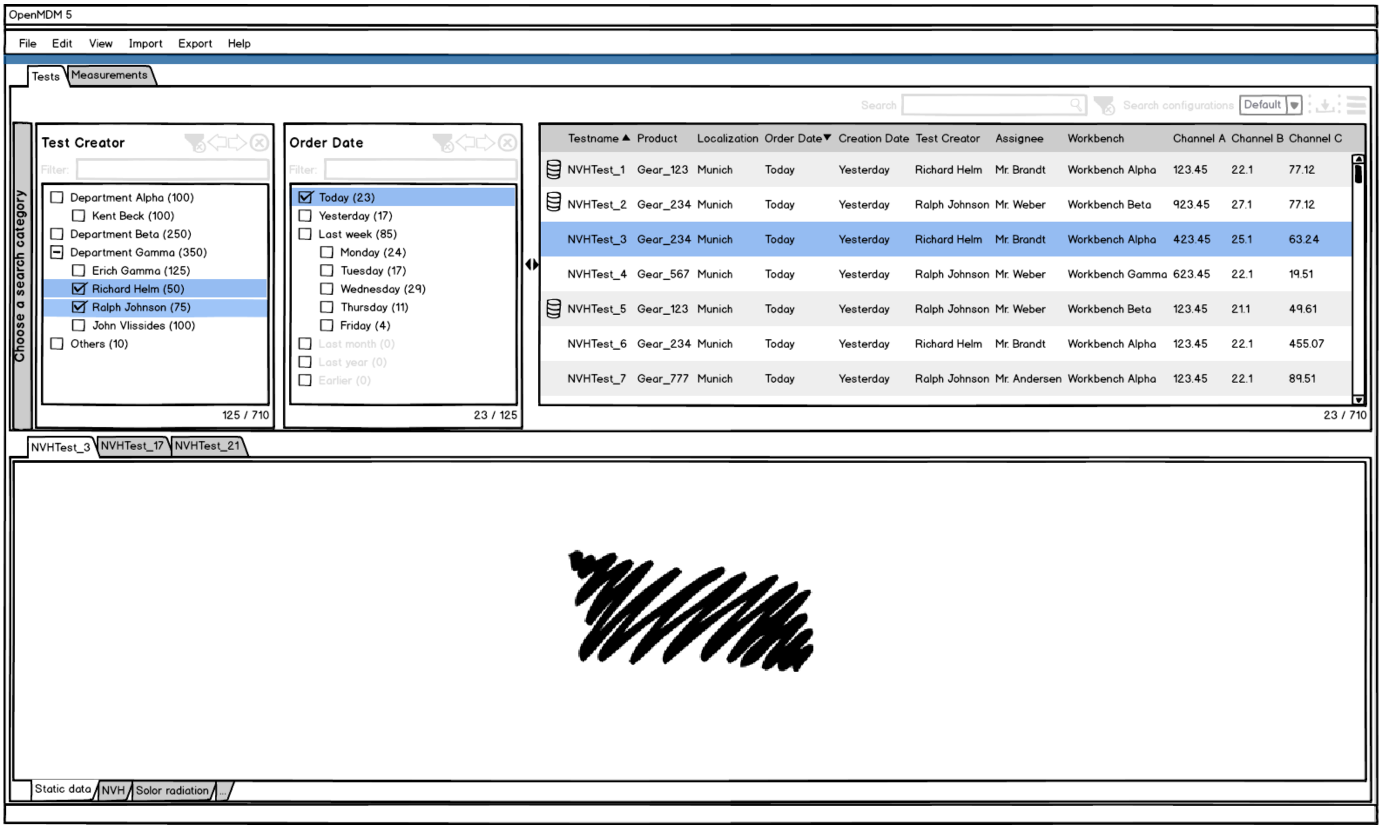
Another toolbar button allows to export the result table content. It is desired to allow different data formats such as ATFX.

Finally an options button hides all rarely used features like managing or publishing saved layouts for a user group.

3.4. Outlook

The result table will provide a filter row (right below the table header) so that the table can be filtered according to each column separately with type specific filter widgets. For example a column displaying date values will provide a filter widget allowing to enter a time period which the data sets have to meet in order to be kept in the table. The specification of those widgets is out of scope of this document.

Via double click on a row in the table a separate details view for this data set is opened. If such a view is already opened for the data set then it is moved to the front. This makes the history functionality obsolete. For all data sets which are of interest detail views can be opened which themselves serve as bookmarks.



Test Creator

- ☐ Department Alpha (100)
  - ☐ Kent Beck (100)
- ☐ Department Beta (250)
- ⊟ Department Gamma (350)
  - ☐ Erich Gamma (125)
  - ☑ Richard Helm (50)
  - ☑ Ralph Johnson (75)
  - ☐ John Vlissides (100)
- ☐ Others (10)

125 / 710

Order Date

- ☑ Today (23)
- ☐ Yesterday (17)
- ☐ Last week (85)
  - ☐ Monday (24)
  - ☐ Tuesday (17)
  - ☐ Wednesday (29)
  - ☐ Thursday (11)
  - ☐ Friday (4)
- ☐ Last month (0)
- ☐ Last year (0)
- ☐ Earlier (0)

23 / 125

Testname ▲	Product	Localization	Order Date ▼	Creation Date	Test Creator	Assignee	Workbench	Channel A	Channel B	Channel C
NVHTest_1	Gear_123	Munich	Today	Yesterday	Richard Helm	Mr. Brandt	Workbench Alpha	123.45	22.1	77.12
NVHTest_2	Gear_234	Munich	Today	Yesterday	Ralph Johnson	Mr. Weber	Workbench Beta	923.45	27.1	77.12
NVHTest_3	Gear_234	Munich	Today	Yesterday	Richard Helm	Mr. Brandt	Workbench Alpha	423.45	25.1	63.24
NVHTest_4	Gear_567	Munich	Today	Yesterday	Ralph Johnson	Mr. Weber	Workbench Gamma	623.45	22.1	19.51
NVHTest_5	Gear_123	Munich	Today	Yesterday	Ralph Johnson	Mr. Weber	Workbench Beta	123.45	21.1	49.61
NVHTest_6	Gear_234	Munich	Today	Yesterday	Richard Helm	Mr. Brandt	Workbench Alpha	123.45	22.1	455.07
NVHTest_7	Gear_777	Munich	Today	Yesterday	Ralph Johnson	Mr. Andersen	Workbench Alpha	123.45	22.1	89.51

23 / 710

Figure 12. Separate detail views can be opened for each data set

When closing the application the currently applied facet component and table composition shall be stored and reapplied on the next restart.

The facet filter mechanism can be enhanced even more by supporting OR, AND and NOT concatenation of facet items within one facet. Multiple visualizations and interaction concepts are thinkable, such as:

- The user clicks on a toggle button or uses a drop down menu within the facet to express which type of concatenation shall be applied.
- Multi state checkboxes in front of every search category value (= facet item) can be used to toggle to the the desired concatenation type.

3.5. Benefits

The new concept implements the strengths of the navigator and search module of the current application, omits their weaknesses and moreover provides several benefits:

One interaction pattern can be used for both exploration of and search for data. Thereby all possible values with their occurrences are presented to the user immediately and still he can drill down and filter for a special entry. The user will never have to guess which data formats apply.

Furthermore outliers and missing data values can be identified very quickly. As indicated special facet items are reserved for this.

As each facet can provide a tree view of facet items, the user can combine as many 'navigators' as he wishes. He only needs to add further facets if he needs them to shrink down the number of items in the result table.

As outlined above, the parts of the user interface currently not needed are made transparent - they are still one mouse click away but do not distract the user from his goal. Other, less often used functionalities are hidden behind option buttons.

Facet item trees inside the facets will not get very large and are sorted alphabetically by default so that there is no need for a sort by type or creation date.

Wildcards are supported, but not enforced.

As the amounts of matching data sets are indicated with every facet item the facet component also allows statistical analysis. Managers could use the OpenMDM application to evaluate e.g. which tests were executed on a certain unit, which unit was involved in which test project, which tests were created but not yet ordered by colleagues XYZ or UVW, how many tests were executed on workbench ABC today, how many tests are included in project MNO or how many tests are ordered but not yet executed.

Chapter 4. Appendix

4.1. Demo video

4.2. Mockups

4.2.1. Facet

4.2.2. Facet chooser

4.2.3. Facet Component

4.2.4. Search Configuration Chooser

4.2.5. Facet Component Integration with collapsed Facet Chooser

4.2.6. Facet Component Integration with expanded Facet Chooser

4.2.7. Facet Component Integration with collapsed facet component