1. Motivation

We all know how zoom in / zoom out works on mapping platforms like google maps, maps.cz etc. Zoom is used to increase or decrease the zoom level at a specific point and show more or less detail on the map.

The extension uses the same approach, namely, when you zoom in, you see more detail in terms of nodes, and when you zoom out, you see less detail in terms of nodes.

The main approach of the extension is to make large graphs more user-friendly, namely more readable and understandable.

1. How it works

The "Grouping of clusters" algorithm first clusters the nodes into a cluster, and then collapses that cluster into a single group node. The clustering of nodes is determined based on the hierarchical class, the parent node, the level of the hierarchy in which the node resides, and the visual class.

1. Contributions

The visual configuration is extended with visual layout constraints which can be used to restrict the way the knowledge graph is visualized. To support visual layout constraints, we extend the Knowledge Graph Visual Browser ontology with new terms.

Figure 1 shows the extension of the ontology as a UML class diagram.

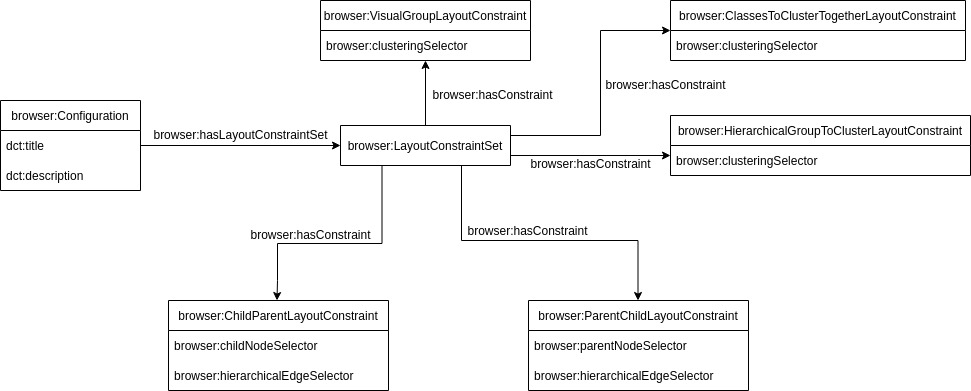


Figure 1. Extended Knowledge Graph Visual Browser ontology for defining extended visual configurations.

In the visual configuration we extend every entity node with additional hierarchical and visual group classes, representing hierarchical and visual groups to which the node belongs, respectively.

A visual group is a cluster of nodes assigned to the same visual group class and placed under the same pseudo-parent node.

A hierarchical group is a cluster of nodes related to each other by parent-child relationships and together forming a hierarchy. In a hierarchy, relationships are not represented as a line, but in such a way that the child node is placed inside the parent node.

Expansion and preview queries defined in the visual configuration are extended with a new variable **?groupclass** that is bound to the visual class representing either the hierarchical or visual group class. It is common to consider a hierarchical class as a group class as well, because we can interpret a root ancestor node in a hierarchy as a pseudo-parent node.

The visual knowledge graph is associated with two new functions and that map each entity node to a set of literals called visual group classes and hierarchical group classes, respectively. Entity node is placed under the visual group of class defined by . And similarly, entity node is placed in the hierarchical group of class defined by . It is also possible that the node is not assigned to a visual group class or a hierarchical group class.

A visual configuration extended with visual layout constraints is bounded to a concrete . As for now, the Knowledge Graph Visual browser supports only “hierarchical parent-child or child-parent", “hierarchical groups to cluster”, “visual group” and “classes to cluster together” layout constraints.

The “parent-child” (respectively “child-parent”) visual layout constraint defines the visual class of the node that plays the role of the parent (respectively child) node and the visual class of the edge, which should be rendered as a hierarchical transition between the parent and the child rather than a line. The visual layout constraint is expressed as an instance of the browser:ParentChildLayoutConstraint (respectively browser:ChildParentLayoutConstraint) class. The visual class of the parent node is assigned to this class with the browser:parentNodeSelector (respectively browser:childNodeSelector) property. The visual class of the edge is assigned to the class with the browser:hierarchicalEdgeSelector property.

The “hierarchical groups to cluster” visual layout constraint defines the visual class, namely hierarchical group class, that represents a hierarchical group. All nodes within such a hierarchical group are allowed to be clustered and grouped. All other nodes within other hierarchical groups cannot be clustered and grouped by the extension algorithm. The visual layout constraint is expressed as an instance of the browser:HierarchicalGroupToClusterLayoutConstraint class. The hierarchical group class is assigned to this class by the browser:clusteringSelector property.

The “visual group” visual layout constraint defines the visual class, namely visual group class, that represents a visual group. The visual layout constraint is expressed as an instance of the browser:VisualGroupLayoutConstraint class. The visual group class is assigned to this class by the browser:clusteringSelector property.

By default, grouping of clusters algorithm clusters nodes that only have the same visual class. The “Classes to cluster together” visual layout constraint defines a set of visual classes that can be clustered together and then grouped into one group. The visual layout constraint is expressed as an instance of the browser:ClassesToClusterTogetherLayoutConstraint class. Each visual class from a set is assigned to this class by the browser:clusteringSelector property.

The stateless server is extended with a new request handler that prepares visual layout constraints and sends them to the client, which then uses them to change how the knowledge graph is visualized.

1. Example

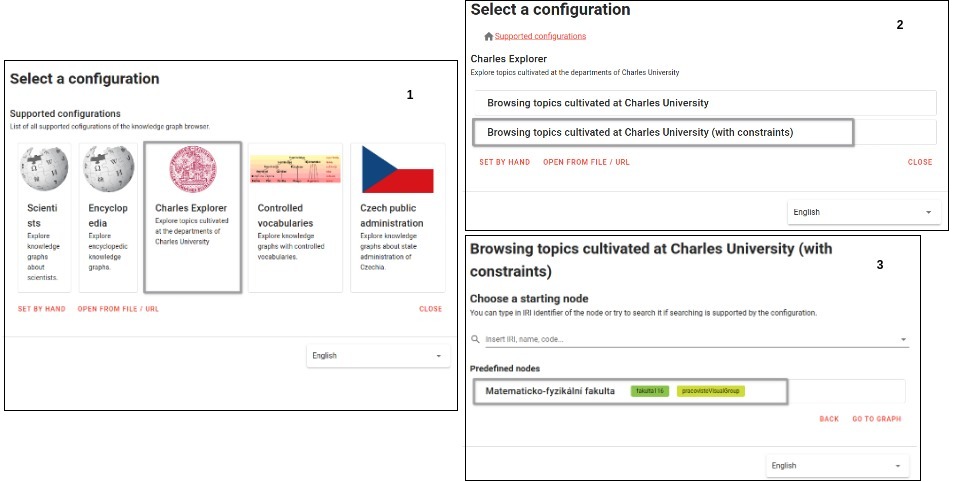


Figure 2. Choice of the configuration and starting node (screenshots 1-3).

Let us demonstrate our extension on a visual knowledge graph about Matematicko-fyzikální fakulta.

At the beginning, the user chooses the Charles Explorer meta-configuration (Figure 2, screenshot 1). The user then chooses the configuration supporting the visual layout constraints (Figure 2, screenshot 2). Not all the configurations support visual layout constraints. The user then chooses the starting node from the list of starting nodes, in our case it is Matematicko-fyzikální fakulta (Figure 2, screenshot 3). The client then visualizes the selected starting node and reads layout constraints from the server. The rest of the visualization part is as usual.

An interesting part happens when the user starts exploring a graph. When the user selects the node on the graph, the client loads the node’s preview. If the node has assigned a hierarchical class (in the preview query) representing a hierarchical group that is allowed to be clustered and grouped, the client shows this class labeled as hierarchical class in the preview section on the side panel (Figure 3 screenshot 1).

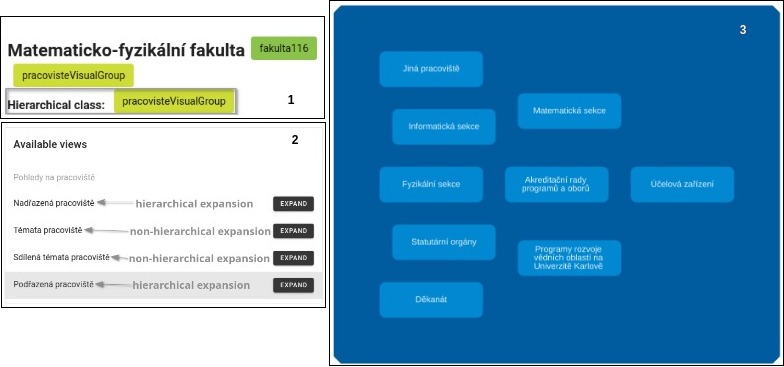


Figure 3. A possible user’s scenario of exploring the knowledge graph with Matematicko-fyzikální fakulta in KGBrowser (screenshots 1-3).

Expansions that expand the node with its neighborhood using hierarchical relationships are hierarchical expansions. There are hierarchical and non-hierarchical expansions (Figure 3, screenshot 2).

The user selects the Podřazená pracoviště view and clicks the Expand button (Figure 3, screenshot 3). The client then performs the expansion action. Since the expansion is hierarchical, the client connects the neighborhood of the node using hierarchical relationships (Figure 3, screenshot 3). Here the Matematicko-fyzikální fakulta node represents a parent node.

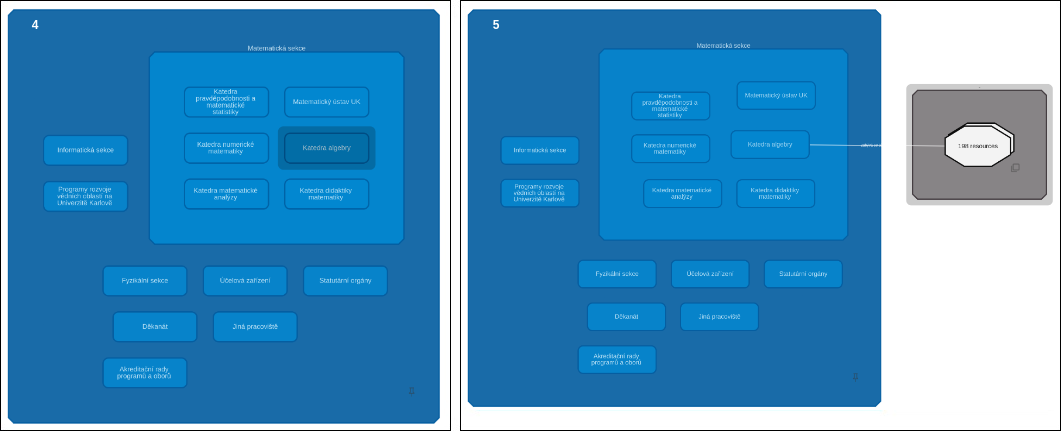


Figure 4. A possible user’s scenario of exploring the knowledge graph with Matematicko-fyzikální fakulta in KGBrowser (screenshots 4-5).

The user then selects the Matematicka sekce node, clicks Podřazená pracoviště expand button in available views and selects Katedra algebry node (Figure 4, screenshot 4).

The user then selects the Témata pracoviště view of the Katedra algebry node and clicks the Expand button. Now, the expansion is non-hierarchical, so the client connects the expanded neighborhood with a visual line (Figure 4, screenshot 5).

This way, the user opens a group of nodes having tema visual class. The visual configuration specifies nodes assigned to tema visual class as a visual group. Therefore, the client creates the pseudo-parent node representing the tema visual group (Figure 4, screenshot 5, right side).

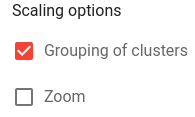


Figure 5. Scaling options. The user selects “Grouping of clusters” option.

The user then decides to utilize the “Grouping of clusters” technique which brings the zoom approach used in mapping platforms to the Knowledge Graph Visual Browser. The user selects the “Grouping of clusters” option in the checkbox (Figure 5) and clicks the “minus” button. The client then groups nodes on the lowest (deepest) hierarchical level shown on the graph (Figure 6, screenshot 6). Then the user decides to go further and clicks multiple times the “minus” button. After a few iterations, the user gets the result shown on Figure 6, screenshot 7. The user clicks the “minus” button once again, and at this time the client collapses the child group into its Matematicka sekce parent node. After a few more iterations with the “minus” button, the user gets to the state where the graph cannot be collapsed further (Figure 6, screenshot 9). This state represents the highest abstract level of the hierarchy. In this state, the graph shows the least amount of detail possible.

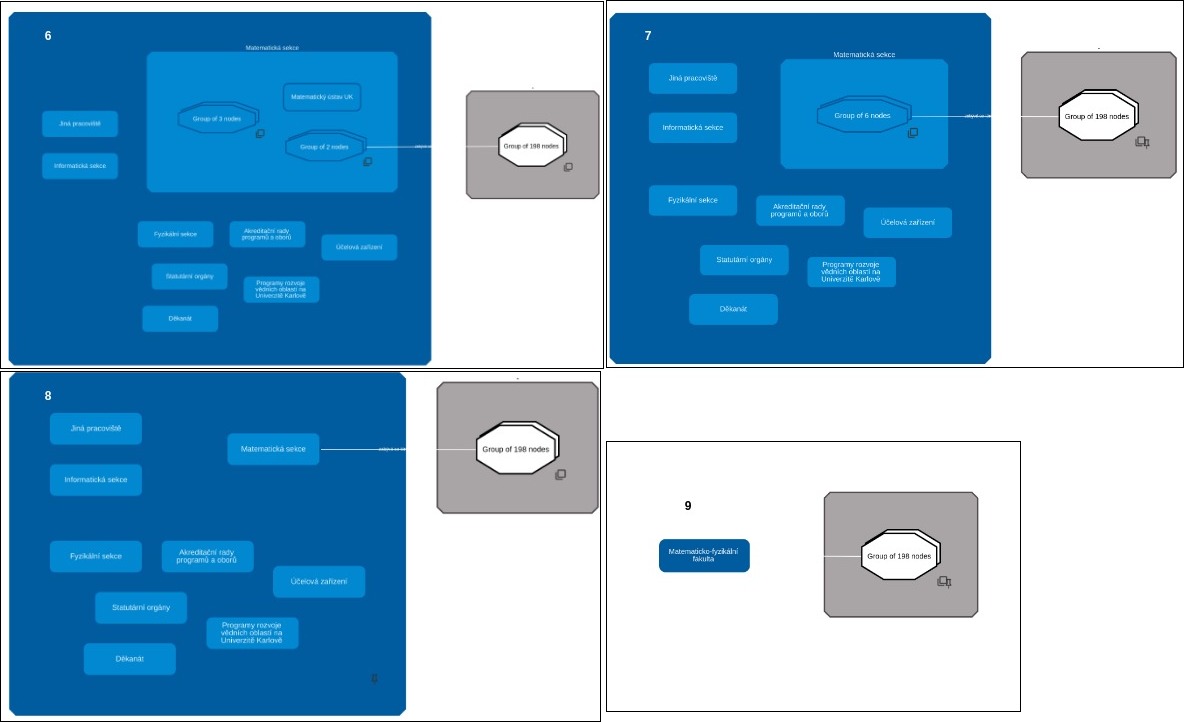


Figure 6. A possible user’s scenario of exploring the knowledge graph with Matematicko-fyzikální fakulta in KGBrowser (screenshots 6-9).

The user then decides to click the “plus” button multiple times. With each iteration, the user gradually returns to the state from which he started (Figure 4, screenshot 5).