IMiGEr raw input format

Tomáš Šimandl

September 2018

1 Introduction

Interactive Multimodal Graph Explorer(IMiGEr) is used for visualization of diagrams which are defined in JSON file which will be described in this document. This JSON contains data about diagram and actual state of diagram in application (decomposition, selected vertex, vertices in side bar, etc.).

2 Format description

JSON is composed from several objects which will be described in this section. Complete JSON format is in appendix A.

2.1 attributeTypes

In this array are defined all possible attributes that can be used in array `attributes` which is defined in `vertices` (section 2.4) and `edges` (section 2.5) arrays. In `attributes` arrays are used only indices to items in `attributeTypes` array. First item has index zero. Attribute `dataType` contains data type of attribute (number, string, enum, etc.). Attribute `name` is name which will be displayed to user in application. `text` is not used now and can be set to empty string or used as description of attribute.

```
"attributeTypes": [
    {
        "dataType": <string>,
        "name": <string>,
        "text": <string>
    },
    ...
],
```

2.2 edgeArchetypes

Array contains all possible archetypes of edges. In edge (section 2.5) is only index to this array. First item has index zero. Attribute `text` is not used and can be set to empty string or can be used as description of archetype.

```
"edgeArchetypes": [
    {
        "name": <string>,
        "text": <string>
    },
    ...
],
```

2.3 vertexArchetype

Array contains all possible archetypes of vertices. In vertex (section 2.4) is only index to this array. First item has index zero. Attribute `icon` contains `svg` description of icon which will be displayed in vertex with given archetype. `Svg` must contains only object defined in rectangle from [0;0] to [12;15]. `text` is not used and can be set to empty string or can be used as description of archetype.

```
"vertexArchetypes": [
    {
        "icon": <string>,
        "name": <string>,
        "text": <string>
    },
    ...
],
```

2.4 vertices

Contains definition of all vertices in diagram. Attribute `archetype` is only index to `vertexArchetypes` array (section 2.3). Array `attributes` contains arrays of size two. First item of array contains index to `attributeTypes` array (section 2.1) and second item contains value which can be another array or string value. Attribute `id` is very important and should not be changed when is manipulated with diagram. `name` is displayed to user in application. `text` is not used and can be set to empty string or can be used as description of vertex. Last attribute `position` is optional and is used for decomposition of vertex in application. When attribute is not set random decomposition will be used. Position is relative to parent component.

```
"vertices": [
    {
```

```
        "archetype": <int>,
        "attributes": [
            [
                <string-index>
                <string-value>
            ],
            ...
        ],
        "id": <int>,
        "text": <string>,
        "name": <string>,
        "position": {
            "x": <float>,
            "y": <float>
        }
    },
    ...
],
```

2.5 edges

Array contains all edges between vertices in diagram. Because between two vertices can be more than one edge, item contains attribute `subedgeInfo` which contains only attributes which can be different. Attribute `subedgeInfo.id` is identification number of one edge, but `id` contains identification of this group of edges which have equal `from` and `to` attributes. `subedgeInfo.id` should contains original identification which was used in original data format (format from which is diagram converted to this format). Attribute `subedgeInfo.archetype` contains only index to `edgeArchetypes` array (section 2.2). Array `attributes` contains arrays of size two. First item of array is index to `attributeTypes` array (section 2.1) and second item contains value which can be another array or string value. Attributes `from` and `to` contains identification number of vertices which are connected with this edge. `text` is not used and can be set to empty string or can be used as description of this edge.

```
"edges": [
    {
        "subedgeInfo": [
            {
                "id": <int>,
                "archetype": <int>,
                "attributes": [
                    [
                        <string-key>
                        <string-value>
                    ],
```

```
                    ...
                ]
            },
            ...
        ],
        "from": <int>,
        "id": <int>,
        "text": <string>,
        "to": <int>,
    },
    ...
],
```

2.6 possibleEnumValues

This array contains all possible values of `attributes` item in `vertices` (section 2.4) and `edges` (section 2.5) which data type is enum. It is used for filters on application front-end. First attribute is index to `attributeTypes` array (section 2.1) and second attribute is array of strings with possible values.

```
"possibleEnumValues": {
    <attribute idx>: [
        <string>,
        [<string>, ...],
        ...
    ],
    ...
},
```

2.7 groups

Vertices can be grouped to groups. When no group is needed empty array should be used. Attribute `name` will be displayed to user in application. Array `verticesId` contains all identification numbers of vertices which are in this group. Vertex must be only in one group. Last attribute `position` is optional and is used for store of decomposition of group in application. When attribute is not set, random decomposition will be used. Position is relative to parent component.

```
"groups": [
    {
        "id": <int>,
        "name": <string>,
        "verticesId": [<int>, <int>, ...],
        "verticesEdgeFromId": [<int>, <int>, ...],
```

```
        "verticesEdgeToId": [<int>, <int>, ...],
        "position": {
            "x": <float>,
            "y": <float>
        }
    },
    ...
],
```

2.8 sideBar

Array contains definition of elements which are displayed in application in side bar. Attribute `id` is string identification number of vertex or group. Format is `vertex-id` for vertex and `group-id` for group. Attribute `isIconsDisplayed` indicates if symbol will be displayed in neighbours (vertex or group which is connected by edge).

```
"sideBar": [
    {
        "id": <string>,
        "isIconsDisplayed": <bool>
    },
    ...
],
```

2.9 highlightedVertex

This attribute contains identification number of vertex or group which is selected in application. Format of value is `vertex-id` for vertex and `group-id` for group.

```
"highlightedVertex": <string>,
```

2.10 highlightedEdge

Contain identification number of edge which is selected in application. Identification number is not `edge.subedgeInfo.id` but it is `edge.id`.

```
"highlightedEdge": <string>
```

Appendices

A Complete JSON format

```
{
    "attributeTypes": [
        {
            "dataType": <string>,
            "name": <string>,
            "text": <string>
        },
        ...
    ],
    "edgeArchetypes": [
        {
            "name": <string>,
            "text": <string>
        },
        ...
    ],
    "vertexArchetypes": [
        {
            "icon": <string>,
            "name": <string>,
            "text": <string>
        },
        ...
    ],
    "vertices": [
        {
            "archetype": <int>,
            "attributes": [
                [
                    <string-index>
                    <string-value>
                ],
                ...
            ],
            "id": <int>,
            "text": <string>,
            "name": <string>,
            "position": {
                "x": <float>,
                "y": <float>
```

```
            }
        },
        ...
    ],
    "edges": [
        {
            "subedgeInfo": [
                {
                    "id": <int>,
                    "archetype": <int>,
                    "attributes": [
                        [
                            <string-key>
                            <string-value>
                        ],
                        ...
                    ]
                },
                ...
            ],
            "from": <int>,
            "id": <int>,
            "text": <string>,
            "to": <int>,
        },
        ...
    ],
    "possibleEnumValues": {
        <attribute idx>: [
            <string>,
            [<string>, ...],
            ...
        ],
        ...
    },
    "groups": [
        {
            "id": <int>,
            "name": <string>,
            "verticesId": [<int>, <int>, ...],
            "verticesEdgeFromId": [<int>, <int>, ...],
            "verticesEdgeToId": [<int>, <int>, ...]
            "position": {
                "x": <float>,
                "y": <float>
            }
```

```
        },
        ...
    ],
    "sideBar": [
        {
            "id": <string>,
            "isIconsDisplayed": <bool>
        },
        ...
    ],
    "highlightedVertex": <string>,
    "highlightedEdge": <string>
}
```