Visual Representation of the Workflows by Using Sugiyama Algorithm

Huzeyfe Esen
İdea Teknoloji Çözümleri
İstanbul, Turkey
huzeyfe.esen@ideateknoloji.com.tr

Abstract—The whole states and transitions, which are occurred or are possible to be occurred are named as Workflow during a process at any work. The transitions between the states are occurred after related conditions are satisfied. On many companies, these workflows belonging to work processes are stored in database. The tracking of these processes are getting more difficult as the number of the states and the transitions have rose. In order to get this issue solved, the states and the transitions are presented visually in an optimal manner by implementing Sugiyama Algorithm into systems.

Index Terms—Workflow, Graph Theory, Topological Sort, Hierarchical Layout Drawing, Sugiyama Algorithm, MSAGL, Jsplumb

I. INTRODUCTION
Workflow is the name which is used for whole transitions and required conditions at any work process. In Figure 1, various workflows belonging to different processes are depicted. Since these workflows are consisted of 2 layers and 2 states, they can be observed just by tracking the data which is related to them in database.

Fig. 1. Various Workflows Belonging to Different Processes

For some cases, the tracking process cannot be that much trivial. In our system, to carry out tracking a workflow in database, 7 different tables must be observed:

- T_WORKFLOW
- T_WORKFLOW_OPERATION
- T_WORKFLOW_OPERATION_VALID
- T_WORKFLOW_OPR_VALID_VALUES
- T_WORKFLOW_OPERATION_AUTH
- T_SYS_TABLE_COLUMN
- FROM T_LOOKUP_VALUE

Figure 2 depicts a capture from just a part of a table in database. As the number of states and transitions increase, the level of difficulty is affected. Visual presentation of the workflows will play a part in solving this issue. Instead of tracking the tables and columns related to the flow in database, observing the flows presented visually will contribute the user to go through an easier process.

Fig. 2. A Capture from Data related to Workflow in Database

Even if it seems as though visually presentation of workflow is just a drawing process by linking the states to each other based on the connection rules, the case is more complicated contrary to popular belief. The locations of the drawn nodes referring to the states and the conflicts of the drawn connections referring to the transitions between the nodes can cause the workflow to be messed up. Optimizing the locations of the nodes will improve the visibility of the layout in observable manner. Even if Fig. 3 and 4 are not in a hierarchical form, they set good examples to clarify the mentioned optimization. Even though the both layouts depicted in Figures consist of the same transitions and states, it can be easily observed that there is obvious distinction visually between them. It is derived that the localization of the states is the key to presenting the layout in
Localization of the states should be based on some related algorithms in order for the layout to be drawn optimally.

As for handling drawings as seen in Fig. 1, different algorithms should be employed. Topological sort is the very algorithm which we have come across after a detailed research related to that type of drawings.

The states (vertices) are ordered linearly under this topic. It is described that for each $uv$ edge (transition) which is from $u$ vertex towards $v$ vertex, each $u$ vertex comes first in terms of layering (See Fig. 5). If and only if the graph does not involve cycle, the graph can be layered as described above [4].

Topological sorting is used for arranging the vertices layer by layer as mentioned. Optimality is not concerned for the vertices inside the same layer. Another solutions should be looked for in order to arrange the states horizontally. Another issue for the graph is being acyclic. As cyclics usually exist for the flows, other methods which also involve handling this issue will be examined.

III. PROPOSED METHOD

As we have kept researching, we come across hierarchical layout drawing algorithms. It is observed that this type of drawing provides us a structure in the flow type which we are looking for (See Fig. 6).

In Introduction Section, the motivation of the study, aims and information related to scope are stated. In Section 2, literature researches are briefly examined. While in Section 3 the proposed method is explained, in Section 4 the experimental results are focused on based on the implementation of the proposed method. In the last section, the results maintained from the study are summarized.
This type of drawings are also called as ‘Sugiyama-style graph’.

A. Sugiyama Algorithm

As Kozo Sugiyama is the first person who has developed the drawing style like Fig. 6. The algorithm that he has developed is known as Sugiyama [6].

This algorithm is not just only used for finding optimal layer levels of the states, but also is used for detecting optimal locations of them inside the optimal layers. Thus minimum edge confliction is satisfied when the transitions between the states are drawn.

Ideal form in hierarchical drawings is obtained by the relevant directed edges with non-confliction.

In implementation of the algorithm firstly it is checked whether cyclics involve or not in graph. If there exist, it is aimed to break these cycles by changing its structure. In order to carry out this, the edges causing cyclics are directed in reverse direction. These edges are chosen in an optimal way so that the number of them are kept at minimum. Detecting the edges which are to be reversed is NP-complete feedback arc set problem and usually solved by greedy heuristics [7]-[10]. Exact solution is obtained by integer programming. Afterwards vertices are assigned to the related layer and it is forced that the number of the layers are kept at minimum. When localizing horizontally, virtual vertices are created in the related layer. These vertices are omitted and their purpose is to provide minimum conflictions of the edges.

As it is explained above, drawing of the workflow cannot be carried out by localizing the states randomly. Various algorithms should be implemented in order to present it visually in optimal way. In our research, Sugiyama Algorithm is cut out for our desired workflow drawing.

B. Implementation of Sugiyama Algorithm and Drawing of the Workflow

Some ready-to-use libraries exist related to the mentioned algorithms. By just feeding state and transition data into these libraries, promising results can be obtained. Graphiz [11], Microsoft Automatic Graph Layout [12], yWorks [13] and Tulip [14] are a few examples to them.

In order for the states and transitions to be drawn on screen, svg and canvas features of HTML5 seem to be applicable. Besides, there are some ready-to-use libraries like Jsplumb [15], D3 [16] and Raphael [17].

IV. EXPERIMENTAL RESULTS

In this section, the results will be shared based on the steps which are touched on in the previous section.

A. Localization of the States by Employing Sugiyama Algorithm

Detecting the cyclics inside the workflow, reversing the related transitions and implementation of Sugiyama Algorithm are carried out by a library belonging to Microsoft.

In order to be able to proceed in a more efficient and faster way, ready-to-use libraries and tools have been researched. We came across Microsoft Automatic Graph Layout (MSAGL). Even if it is accessible free of charge, for commercial usage it is required to pay an amount of money. As our company is msdn subscriber, we are allowed to use it commercially free of charge.

Before drawing the workflow, data related to states and transitions are converted into a correct format. Afterwards related method is called and using MSAGL [12] optimal locations of the states are obtained.

B. Visually Presentation of Workflow States and Transitions

After the optimal localization process of the states has been carried out by Sugiyama Algorithm which is involved in MSAGL, states and transitions are drawn by using Jsplumb [15] library.

C. Visually Presentatiion of Workflows

First of all, the data related to workflow is retrieved from database by relevant select queries. After these data are processed by Sugiyama Algorithm, the layer order of each state is detected. In addition to that, the states inside these layers are localized. Afterwards, based on these locations, drawing of the workflow is achieved. Examples related to these drawings are seen in Fig. 7 and Fig. 8.

For a transition from a state to another one to be carried out, some conditions must be satisfied. Firstly everyone is not authorized for any transition. Secondly some pre-defined constraints must be met for that transition to be occurred. On our implementation, when the desired transition is clicked, all related conditions are shown to the user on a pop-up window.
V. CONCLUSION

In order to be able to track the states and transitions in a workflow, the data needs to be observed one by one in database. This process does not generate any issue as long as the number of the states and transitions are reasonable. However, as the number of them expands, tracking is getting more difficult. In order to prevent this kind of difficulty, visually presentation of workflow is studied.

Visualization of the states and transitions are as is the case with tracking in database. The states and transitions cannot be observed due to the complexity as the number of them rises. Regarding this issue, Sugiyama Algorithm is implemented for getting optimal solution.

MSAGL library of Microsoft is made use of in implementation of Sugiyama Algorithm. Visualizing the states and transitions are carried out by adopting Jsplumb javascript library after localization of the states. Hence the user has the opportunity for observing workflows in most efficient way.

ACKNOWLEDGMENT

This work was supported by "Idea Teknoloji Çözümleri Bilgisayar Sanayi ve Tic. A.Ş.".

REFERENCES