Speech Enabled Ontology Graph Navigation and Editing

Dimitris Spiliotopoulos[1(✉)], Athanasios Dalianis[2], and Dimitris Koryzis[3]

[1] Distributed Computing Systems, Institute of Computer Science Foundation for Research and Technology – Hellas, Heraklion, Greece
dspiliot@ics.forth.gr
[2] Innovation Lab, Athens Technology Centre, Athens, Greece
t.dalianis@atc.gr
[3] Hellenic Parliament, Athens, Greece
dkoryzis@parliament.gr

Abstract. Graphs are commonly used to represent multiple relations between many items. Ontology graphs implement the connections and constraints between levels of interdependence between nodes; the nodes themselves being the members of the data types. As part of a design-for-all approach, this paper reports on the use of speech for ontology graph navigation and editing. The graphs can be fully created by using voice commands only, essentially creating large and complex ontologies by speech. The formative usability evaluation and user involvement experimentation results revealed that the introduction of speech, greatly enhanced specific parts of the navigation and improved the speed of editing, especially for the trivial, yet time consuming tasks of editing large and complex graphs.

Keywords: Speech · Ontologies · Graph editing · User interface design

1 Introduction

Graphical representation of complex relations between items has been used in abundance in the recent years. Social graphs, in particular, may result in very large structures that deploy techniques such as zoom and pan and instant search for users to be able to browse effectively [1, 2]. Ontology graph is one of several ways of authoring and browsing ontologies, from a range that spans from list, trees and tables to 3D representations [3]. To ensure the visibility of the relations between the entities and the visual recognition of clusters, graphs are opted as an optimal means to visualise for almost all (small to very large) representations.

Recently, graphs have been used as part of advanced web interfaces that were designed for authoring complex ontology applications such as policy modelling [4]. As the graphs become large, problem arise for users that need to view specific entities or clusters. Depending on the size and complexity, ontology graphs may become too hard to follow, especially during the authoring of the ontology itself. Taking a few steps back, the new problem becomes proportionally larger as the size of the graph grows. In

M. Antona and C. Stephanidis (Eds.): UAHCI 2015, Part I, LNCS 9175, pp. 487–494, 2015.
DOI: 10.1007/978-3-319-20678-3_47

application-specific approaches like the one mentioned before, nodes have names that can be as large as sentences. Adding new nodes and relations becomes cumbersome even when the graphs are medium sized, as in Fig. 1.

This work implements and evaluates a speech-enabled navigation and editing approach to enhance the user experience of authors of complex ontology graphs. The following sections present the design rationale and requirements, the set of speech commands that were implemented and the evaluation of the speech based interface compared as part of a new two-modal solution from the initial traditional web interface.

2 Design Considerations

For our design, an existing web interface that was designed to author ontology graphs was used [4]. The aim of the web authoring interface was to enable non-technically proficient authors from diverse work environments (parliamentary assistants, policy makers, crowdsourcing private sector, students) to create domains and policy models with the data that will drive the collection of documents from news pages and social media (Facebook, Tweeter), the sentiment analysis of the collected data sets and the argument extraction. That information is then fed back to the authoring environment for the fine-tuning and later extension of the models [5].

Figure 1 depicts a typical policy model authored and viewed on the aforementioned web interface.

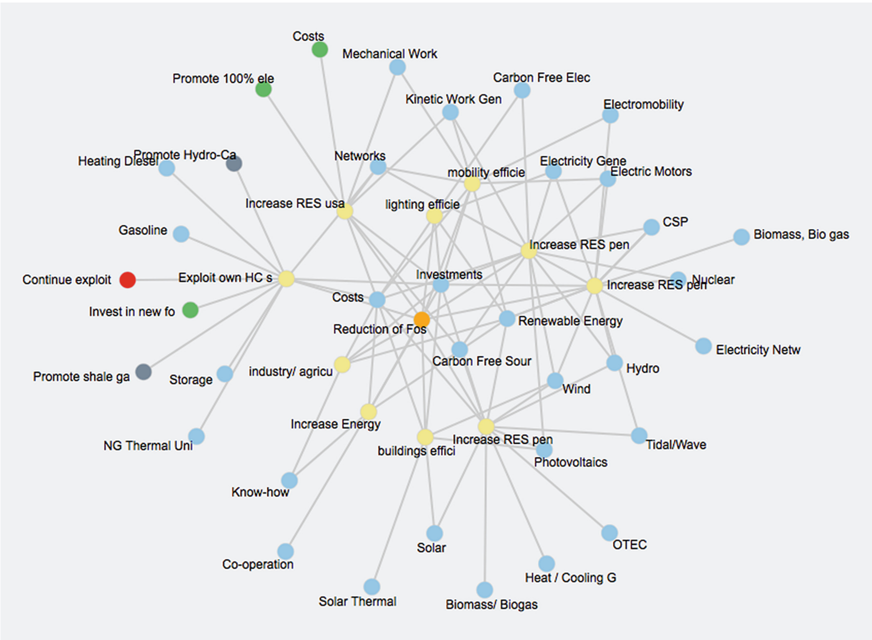

Fig. 1. Policy model ontology graph

The authoring of a policy domain or model is through the same generic concept. The author specifies the ontology domains by adding and editing instances of entities, norms and arguments. These can be connected as to describe the relations between them, essentially forming a graph. The simplest form for a vary small domain or model is a tree. The aim of the web interface was to provide a seamless user experience to the end users, yet enable them to create the envisioned ontology models. The high-level requirements were selected from groups of users from crowdsourcing service provision organizations and political bodies. The contextual framework for the interface specifications has been identified and described by a list of policy model domain specific items. The items include entities, sentiment and opinions, social and demographic information, sentence level arguments from a range of traditional web and social media-related sources, such as Blogs, Wikis, and Social Networks, namely Twitter and Facebook.

The described web interface and authoring approach work very well, utilizing the freedom of relation visualization of graphs to represent ontological structures like policy models and domains. Specific techniques for graph visualization were deployed in order to aide the users, such as zooming in/out and fast centering, panning, highlighting neighbouring nodes on node selection (Fig. 2). Additional non-graph related issues such as the large node names were addressed by displaying the first 16 characters of each node name.

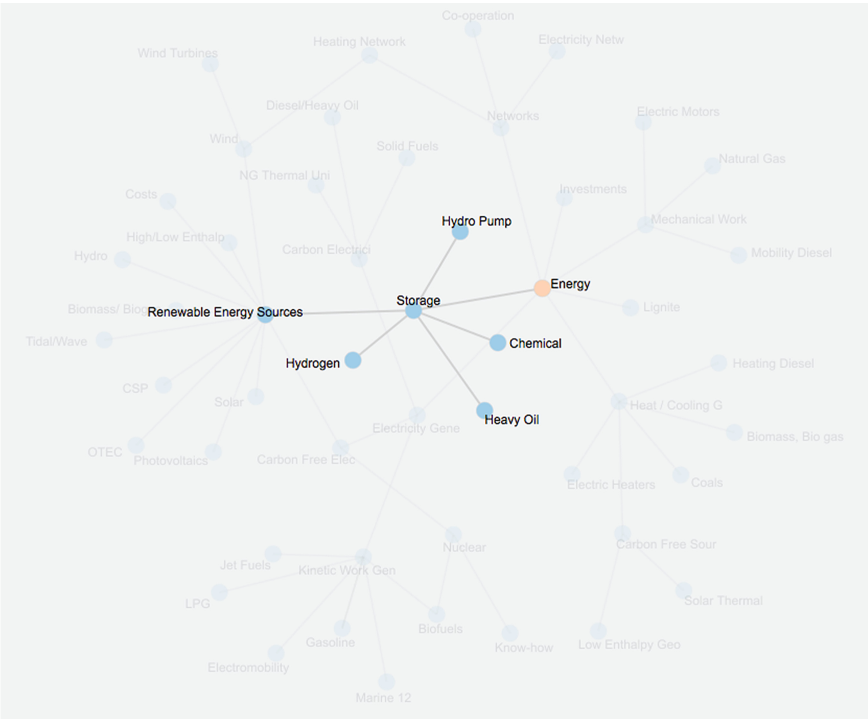

Fig. 2. Highlighting a node and directly connected nodes

However, as the authors progressed and created very large graphs, they reported increasing difficulty finding the node they wanted to edit and clicking to it. Focus group discussion of issues during the next round of design revealed usability issues that directly relate to accessibility. This was evident also from previous studies that explored usability and accessibility as part of the design-for-all methodology for designing voice user interfaces [6].

3 Speech Interface for Graph Editing and Browsing

To address the usability issues above, the second round of the iterative design included the decision to utilize state-of-the-art web speech synthesis and recognition [7, 8] in order to improve the user experience with the ultimate aim to be able to provide a fully speech-driven interface by the end of the lifecycle.

A set of voice commands was implemented over the functionalities of the web interface in order to allow multimodal input to the system. All possible actions that the policy model/domain ontology authors may perform were matched by the voice interface. Two types of input were designed, the commands that initiate content-free interaction with the interface and the ones that include actual content of the model/domain, such as the title text of nodes. A slightly different look into the type of interaction would be to categorize the input as (i) browsing/navigation functionalities and (ii) editing/authoring functionalities. Speech recognition accuracy was more challenging for the latter types of speech commands. Table 1 lists all the speech commands as well as their description. The descriptions, where needed, refer to the non-voice interface interaction for the purpose of direct comparison for the reader.

4 Experiments

Three distinct experiments based on the initial information derived from the user requirements and the web interface prior evaluation round were set up. The purpose was to ensure that the design-for-all approach could integrate with the speech enablement and refine the navigation and editing processes in order to maximize the user engagement and experience. Ten participants (age group 25–42) were asked to evaluate the proposed approach. The aim of the first experiment was to evaluate the impact of the speech based interaction for the graph navigation. The users were asked to verbally search for specific domain entities and semantic tags in order to filter and sort specific entities and relations of interest. They were also asked to use the traditional non-speech enabled interface to achieve similar tasks. The second was to investigate how adding new information and editing existing data could align with the user mental impression of how a domain should be created. That task, being user/domain dependent, was achieved by asking the participants to add new information and evaluate later whether their selection and choices were optimal, considering the use of both speech and non-speech actions that they had at their disposal.

The final experiment was the functional and non-functional usability evaluation, involving both domain experts and casual mobile users. One of the main requirements

Table 1. List of voice commands for graph editing

Command	Description
Open domain	Presents the user's domains. Equivalent to the selection of the action "Open Domain" in the model actions menu
Open policy	Presents the user's policies. Equivalent to the selection of the action "Open Policy" in the model actions menu
New domain	Presents to the user the new domain dialog. Equivalent to the selection of the action "New Domain" in the model actions menu
New policy	Presents to the user the new policy dialog. Equivalent to the selection of the action "New Policy" in the model actions menu
Use model	Presents to the user the reusable models window. Equivalent to the selection of the action "Use models" in the model actions menu
Preview *modelName	Presents the graph of the model (domain or policy) matching the "modelName". If there is no model matching the provided name a message is presented, if there is only one the graph of the model is presented and if there are more than one models matched an options dialog is presented The modelName can be a word or a phrase. It is the same as the mouse over the 'bubble' of the model action
Select node *nodeName	Selects a node from the current model graph matched the "nodeName" provided. If there is no node matching the provided name a message is presented, if there is only one match the node is selected and if there are more than one node matches, an options dialog is presented to the user. Same as clicking the node on the graph for the single result
Select option *option	It selects the option number (integer) from an options dialog presented to the user
Close options	It closes the options dialog
New node	Presents a new node dialog. The new node will be connected to the selected node of the graph. Same as double clicking on the canvas after a node is selected
Create	Completes the creation of the new node and closes the add node dialog
Edit node *nodeName	Presents the edit dialog for the node matching the nodeName. If multiple results the options dialog appears. Same as double clicking a graph node for the single result
Update	Completes the update of the node's data and closes the edit dialog.
English *text	It changes the English text of an add or edit dialog
Access level *level	Changes the access level of a domain or policy when the new model or update dialog is open. Values can be 'private' or 'public'

(*Continued*)

Table 1. (*Continued*)

Command	Description
Argument type *type	Changes the argument type of an argument when the new argument or update dialog is open.
Delete node	Presents the deletion confirmation dialog for the selected node
Delete	Completes the deletion of the node and closes the confirmation dialog
Cancel	Cancels current dialog or action
Connect to node *nodeName	Adds a new link between the node selected and the node matching the nodeName. For the single result case, it is the same as clicking a node after another node is selected
Disconnect node *node1 from node *node2	Deletes the link between two nodes matching the node1 name and node2 name. In case of multiple nodes matching the user is presented an options dialog, holding the combinations found. For the single result case it is the same as double clicking on the link
Deactivate voice commands	The system stops accepting voice commands

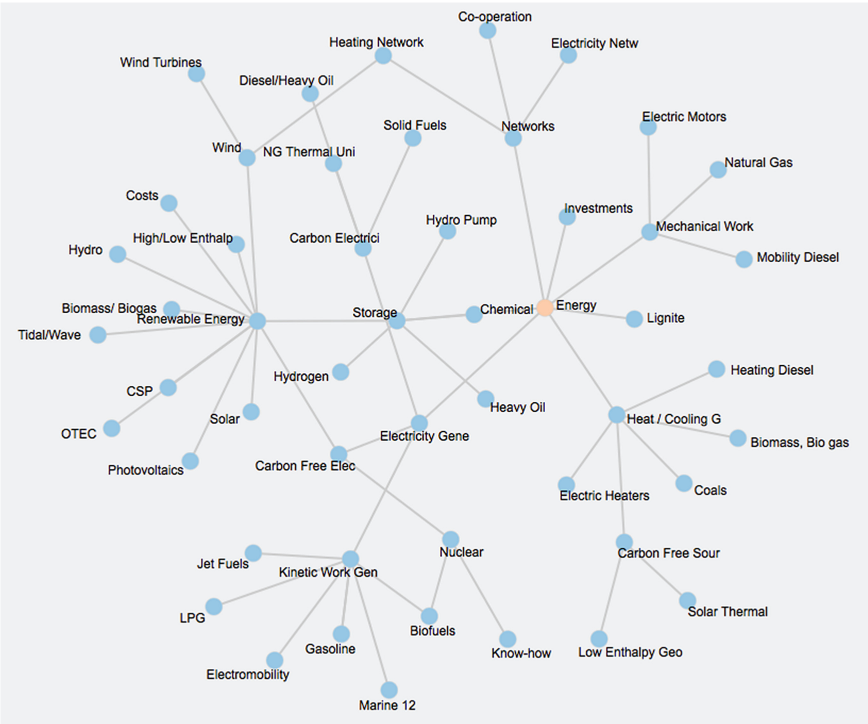

Fig. 3. The test policy domain graph for evaluation

was to measure the impact of the speech driven authoring in terms of time, clarity and acceptance. Figure 3 depicts the test policy domain that the participants were asked to navigate and edit.

5 Evaluation

The participants evaluated the interaction between the traditional non-speech interface and the speech-enabled (Fig. 4). Almost all opted to use speech for the search-related actions expecting to locate the node of interest much faster than by navigating the graph. The overall satisfaction feedback was overwhelmingly favorable for the speech modality, especially for the *find* and *select nodes* actions. The reason was that the voice interface enabled the users to search quickly and center the graph in on their selection. This was particularly apparent for the nodes that had long title text. Editing functions such as the *add* and *delete node/relation* were marginally easier through the use of both modalities, since the users were able to use speech whenever they deemed as an easier path to their goal.

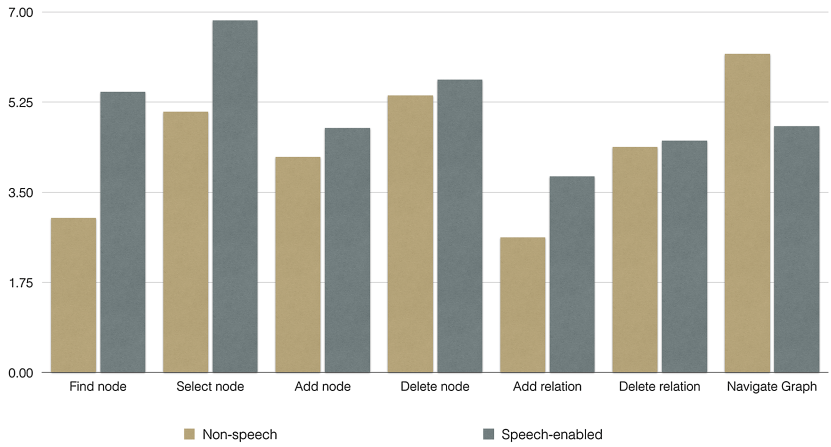

Fig. 4. Evaluation results for non-speech versus speech-enabled interaction

Lastly, the navigation of the graph itself, as a casual browsing task, revealed the shortcomings of the absence of speech commands for the specific generic functionality. No specific commands existed for zooming in/out or panning the graph, hence the users reported that they would have preferred an innovative way to browse, hinting at further research into this method.

6 Discussion

Based on the results of the experimenting with the speech recognition and synthesis tasks, the design of the user interface has been extended to the speech modality that has led to less complexity, as reported by the users. The visual modality was also polished to a more inviting and clear overview of the ontology domain graphs and special features, such as highlighting of the nodes that contain text identified via spoken search, were added. Further work is currently underway for the backend extension of the services that are needed to fully implement the speech web API for the generic graph view functionalities. Additionally, other functionalities that are commonly used in graphs such as dynamic insets [9] may also be implemented into the speech API, essentially allowing the user to preview the insets over the larger graph, while editing. The results of this work are expected to enhance the design of the user interface to support and sustain a multimodal approach to ontology graph authoring.

References

1. Moscovich, T., Chevalier, F., Henry, N., Pietriga, E., Fekete, J.D.: Topology-aware navigation in large networks. In: CHI 2009: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2319–2328. ACM Press, Boston (2009)
2. Rotta, G.C., de Lemos, V.S., da Cunha, A.L.M., Manssour, I.H., Silveira, M.S., Pase, A.F.: Exploring Twitter interactions through visualization techniques: users impressions and new possibilities. In: Kotzé, P., Marsden, G., Lindgaard, G., Wesson, J., Winckler, M. (eds.) INTERACT 2013, Part III. LNCS, vol. 8119, pp. 700–707. Springer, Heidelberg (2013)
3. Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., Giannopoulou, E.: Ontology visualization methods—a survey. ACM Comput. Surv. **39**(4) Article 10 (2007)
4. Spiliotopoulos, D., Dalianis, A., Koryzis, D.: Need driven prototype design for a policy modeling authoring interface. In: Marcus, A. (ed.) DUXU 2014, Part II. LNCS, vol. 8518, pp. 481–487. Springer, Heidelberg (2014)
5. Koryzis, D., Fitsilis, F., Schefbeck, G.: Moderated policy discourse vs. non-moderated crowdsourcing in social networks – a comparative approach. In: Jusletter IT, February 2013, Proceedings of the 16th International Legal Informatics Symposium, IRIS (2013)
6. Kouroupetroglou, G., Spiliotopoulos, D.: Usability methodologies for real-life voice user interfaces. Int. J. Inf. Technol. Web. Eng. **4**(4), 78–94 (2009)
7. Shires, G., Wennborg, H.: W3C web speech API specification, 19 October 2012. https://dvcs.w3.org/hg/speech-api/raw-file/9a0075d25326/speechapi.html. Accessed 29 Jan 2014
8. Annyang Speech Recognition JS Library. https://www.talater.com/annyang/. Accessed 29 Jan 2014
9. Ghani, S., Riche, N.H., Elmqvist, N.: Dynamic insets for context-aware graph navigation. Comput. Graph. Forum **30**(3), 861–870 (2011)