

Analysing a repository of design knowledge with the GO-DKL browser

Andrew Hilts and Eric Yu

Faculty of Information
University of Toronto
140 St. George Street, Toronto, ON
andrew.hilts@utoronto.ca
eric.yu@utoronto.ca

Abstract. This paper presents the GO-DKL browser, a web-based interface designed to support the contextualization and analysis of items within *Goal-oriented Design Knowledge Libraries*. It is still at a very early stage of development, though a small sample of information systems designers evaluated a demonstration of the tool and provided generally positive feedback.

Keywords: Design Knowledge, User Interfaces, Knowledge reuse, Open OME

1 Purpose of the GO-DKL browser

We define a *goal-oriented design knowledge library* as a repository of relational data linking design features with stakeholder objectives. These relationships, as in *i** and related approaches, indicate the contribution type that a child (design feature / task) passes on to a parent (soft goal). Such a repository might be populated with codified empirical data gathered from a review of a domain's literature, as in [1]. The repository database can be queried to retrieve all design features that have an impact on a focal goal, or on all goals in the database, among other possibilities.

The GO-DKL browser is intended to enable system designers to associate situational goals they have elicited and defined with library items, and to subsequently analyze the impact of retrieved design alternatives on those goals. This association is an act of contextualizing the library items; designers can select a subset of knowledge base records that they feel are related to their own project. From there, the system can retrieve design features that contribute to the selected goals.

The GO-DKL browser is intended to facilitate a focused analysis of certain relationships within a goal graph, in an attempt to mitigate the daunting task of assessing complex goal graphs [4]. Specifically, designers can select focal goals, and drill down to reveal their contributing factors (eg: sub goals or design features). Analysts can assess the effects of not including certain design features on high-level goals, or modify the library's relationships based on trusted contextual knowledge.

2 Primary Features

Essentially, the GO-DKL browser is a web-based interface to a MySQL database. This interface is structured to support the analytic processes discussed above, through goal association, browsing, selection, and model exploration and reconfiguration phases.

The model exploration and reconfiguration phases are centered around a tree-list interface generated using JavaScript. The tree-list structure allows relationships to be expanded and collapsed at will so that certain components can be isolated. HTML form elements permit the reconfiguration of contribution relationships. Parent goals are automatically coloured based on their received contributions from child elements, using simplified *i** evaluation concepts, based largely on the propagation rules defined in [2].

The list is populated by the designer's selected goals and any other knowledge base items that contribute to those goals (his or her 'project model'). The actual list presentation is generated by a script that converts tabular records from the knowledge base into hierarchical HTML elements, with high-level goals as parent elements, and design features as child elements¹. Often the same design feature or relationship will appear numerous times throughout the model, in which case the interface treats all instances of a it as one data point.

Therefore, if one instance of a design feature is deselected, all other instances of that feature will be deselected throughout the tree-list. In this manner, if an analyst selects a design feature because of its contribution to a focal goal, and similarly deselects another, he or she may then examine other goals to see the effects of those choice in other parts of the model.

3 Limitations, Evaluation and Future Work

While the tree-list view can help an analyst to focus intently on the contributions to a single goal (which can be difficult to do with complex models), the 'bigger picture' may be harder to assess. To mitigate this limitation, each high-level goal in the tree-list features an option to export a *model slice*[3], which retrieves all design features contributing to that goal, *in addition to* all other goals affected by those design features. In this manner, an analyst may examine the larger context of design decisions within his or her project model. The model slice is dynamically exported as a *Q7 file*, which is interoperable with Open OME².

Several system design practitioners were shown a demonstration of the GO-DKL browser. They generally deemed it to be a valuable way of retrieving and informally assessing the findings of related projects in comparison to their own endeavours, though several usability critiques emerged. Future work would address these criticisms, as well as attempt to better interoperate with other goal modeling tools.

¹ A maximum of 5 levels of decomposition is currently supported.

² Importing Q7 files is currently unsupported.

4 Availability

Source code³ and a high-level description⁴ are currently available; the community is invited to contribute and expand on this work.

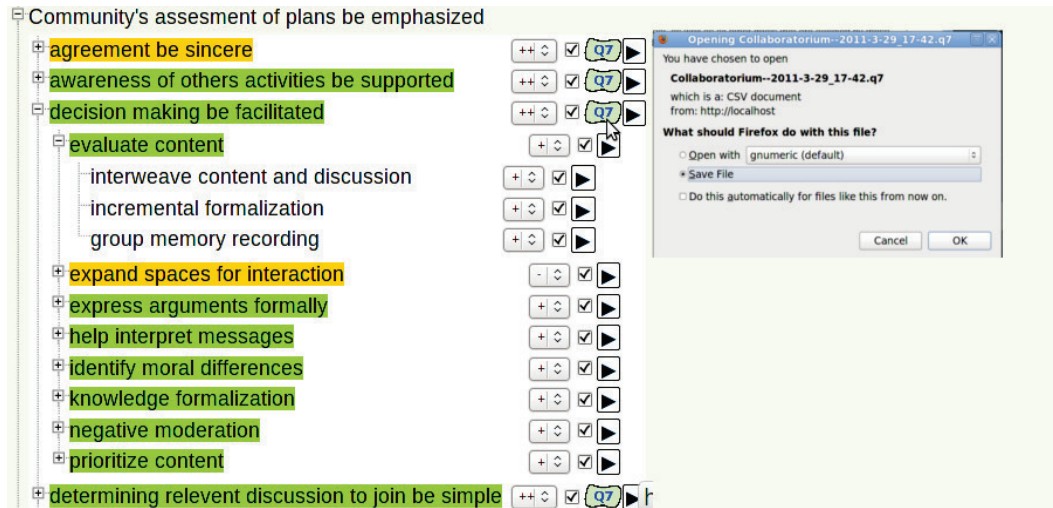


Fig. 1. A 'project model' of selected relationships may be browsed and reconfigured through GO-DKL browser's tree-list interface. Model slices for each high-level goal may be downloaded and imported into Open OME for further analysis.

References

1. Esfahani, H., Yu, E.: A Repository of Agile Method Fragments. In: Münch, J., Yang, Y., Schäfer, W. (eds.) LNCS 6195: New Modeling Concepts for Today's Software Processes. pp. 163–174. Springer, Berlin / Heidelberg (2010)
2. Horkoff, J., Yu, E.: A Qualitative, Interactive Evaluation Procedure for Goal- and Agent-Oriented Models. In: CAISE Forum. CEUR-WS.org, vol. 453. pp. 19–24 (2009)
3. Leica, M.F.: Scalability concepts for i* modelling and analysis. Master's thesis, University of Toronto (2005)
4. Pastor, O., Estrada, H., Martínez, A.: Strengths and Weaknesses of the i* framework. In: Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J. (eds.) Social Modeling for Requirements Engineering. pp. 607–643. MIT Press, Cambridge, MA (2011)

³ <https://github.com/andrewhilts/Go-DKL> also features a detailed README

⁴ <http://www.designknowledge.org>