Adaptive Visualization of Plans

Nava Tintarev, Roman Kutlak, Judith Masthoff,
Kees Van Deemter, Nir Oren, and Wamberto Vasconcelos

Department of Computing Science*
University of Aberdeen
{n.tintarev}@abdn.ac.uk

Abstract. It is often difficult for humans to understand what course of action is proposed in a plan or workflow. This is particularly the case for long plans, or plans with multiple actors. Our contributions are a) the ability to present plans as both text and graphics and b) a method of filtering and highlighting, in both modalities, which focuses the information presentation to the portion of the plan which is relevant to a particular user – i.e., a view based on their roles and capabilities.

Keywords: Adaptation, Visualization, Natural Language

1 Introduction

The output from A.I. planners (e.g. PDDL) and business workflows (e.g. YAWL, BPMN, etc.) can often be difficult for humans to understand. In particular for large plans with multiple actors, it may be difficult for a human to understand what they need to do, or focus on. The aim of the SAsSy project[1] is to reduce the opacity of such plans. To this end, this paper describes ways of adapting the presentation of plans. We show how plans can be presented as either graphics or text (modality adaptation) and how these can be highlighted and filtered to focus on the most relevant information (view adaptation). This paper follows Shneiderman's Information Seeking mantra: *"Overview first, zoom and filter, then details-on-demand."* [1], focusing specifically on "zoom and filter".

We illustrate the functionality of our system using an example from the delivery logistics domain, originally taken from the International Planning Competition[2]. Our sample plan describes how four objects (a truck, a piano, a table and a drum) are delivered to different locations (cities and airports). The plan also contains a number of resources (trucks and airplanes). In addition to being able to present in two modalities (text or graphics), the system can use highlighting and filtering as a means to focus on the most relevant information to

* This research has been carried out within the project "Scrutable Autonomous Systems" (SAsSy), funded by the Engineering and Physical Sciences Research Council (EPSRC, UK), grant ref. EP/J012084/1.

[1] `http://www.scrutable-systems.org/`, retrieved April 2014

[2] `http://ipc.icaps-conference.org/`, retrieved April 2014

a user (given their role and capabilities). The aim is to reduce the amount of information the user needs to deal with, and thereby reduce their cognitive load.

Previous work on visualizing plans has looked at filtering graphs by content [5], and applying fish-eye views to grow or shrink parts of a graph [6]. There is also research on verbalizing plans generated by A.I. planning systems [4]. However, these approaches do not consider that individual differences in user needs, abilities, and preferences can have a large impact on user performance and satisfaction when using visualizations [2].

Our approach differs in that it proposes filtering based on a user model, and introduces a method for including all the required steps, i.e. including dependencies. Our method also works for *both* graphics and natural language. This makes it possible to study the effect of tailoring modality, which may be useful when a user has strong visual working memory, but poor verbal working memory.

2 System Description

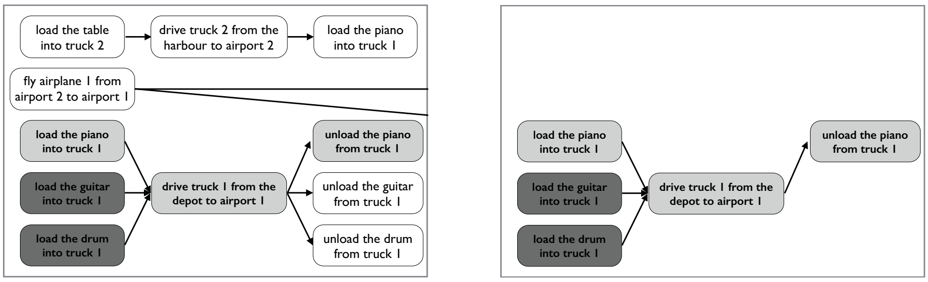
Our system is developed in Python and is available under the BSD license[3]. [7] describes the reasoning in the system, based on argumentation theory, which supports explanation mechanisms. The system can present plans in two modalities (as either graphics or natural language), and can highlight/filter plans based on user views (based on role or capability).

2.1 Modality

Natural Language Generation. (NLG) is the study of algorithms which produce texts in English or other human languages, from non-linguistic representations of information. Instead of presenting a plan as a sequence of tasks as produced by an A.I. planner (e.g., `(load-truck obj12 tru1 pos1), (load-truck obj11 tru1 pos1), (load-truck obj13 tru1 pos1)`) our system presents the plan as text: e.g., *Load the piano, the guitar and the drum into truck 1.*
We use NLG techniques to supply a summary (e.g. *"The workflow has 21 tasks. The workflow has 0 choices. "*). We also use aggregation (combining simple sentences together for better presentation) and referring expression generation (e.g., using pronouns when referring to past entities) to improve the presentation of the full plan.

Graphical presentation. The plan can also be represented as a graph, such as the one in Figure 1. Here, each action is a node, and edges are transitions to other possible actions. For simplification, the example plan in this paper assumes that there are no decisions points – there is only one recommended course of action. In addition, we make the simplifying assumption that parallel actions need to be completed before the next step in the plan can be executed. The algorithm below can be generalized to plans with choice points without too much difficulty.

[3] `https://bitbucket.org/rkutlak/sassy`



(a) ***Highlighting only (cropped).*** (b) ***Filtering and highlighting.***

Fig. 1: Graphical presentation of a plan in the logistics domain. All parallel steps are required for successful completion of the plan. Highlighting for the piano object is applied. Steps which directly involve the piano have a light gray background, while co-dependent steps have a dark gray background.

2.2 View

Algorithm 1

$o \leftarrow$ *selected object*
$T \leftarrow$ *all tasks*
$S \leftarrow \{t \in T | manipulates(t, o)\}$

for $s_1, s_2 \in S$ **do**
  $paths \leftarrow all\_path(s_1, s_2)$
  add tasks in paths to S
end for

for $s \in S$ **do**
  $T' \leftarrow required\_tasks(s)$
  $S \leftarrow S \cup T'$
end for
return S

This section describes an algorithm which aims to focus the information presented to a user using filtering and highlighting. The algorithm can be applied to both text and graphics to focus on those parts of the plan that are most relevant to a given user. Highlighting emphasizes steps that are relevant to the user (Figure 1a), while filtering hides the steps that would not be highlighted, i.e. portions of the plan not relevant to the user (Figure 1b).

Relevance can be determined by the role of someone enacting a plan (e.g. air-traffic controller), or their capabilities (e.g. the person who can operate a fork-lift). For example, a fork-lift operator may only want to know about actions relating to their fork-lift, while an air-traffic controller may only want to know what happens to *all* airplanes (not just one). Note that while the example below is filtered by a particular object (e.g. piano), filtering by object type (e.g. vehicle), or multiple objects is also supported.

Algorithm 1 works as follows: Given an object, for example the piano, the algorithm first selects all tasks that operate on the given object. These are the tasks `load the piano into truck 1` and `unload the piano from truck 1` colored in light gray. The algorithm then finds all paths between each pair of the selected tasks. All tasks on these paths are then added into the list of selected tasks. This corresponds to the task `drive truck 1 from the depot to airport 1` also colored light gray. Lastly, the algorithm inspects all the selected tasks and checks if any of them require completion of other tasks (indicated by multiple arrows arriving at a node in the workflow). In the example, `drive truck 1 from the depot to airport 1` requires the completion of loading all three objects so the two tasks `load the guitar into truck 1` and `load the drum into truck 1` are also selected (colored dark gray).

This example requires all steps to be completed. In a plan with choices, the algorithm includes *all* the paths between a set of actions. An alternative, more aggressive approach, would require a stricter definition of "required" nodes (e.g. include exactly one of the paths).

3 Next Steps

We have introduced a system which can present a plan as both text and graphics. It can also filter and highlight parts of a plan according to areas of relevance based on a user's role or capabilities. The system therefore allows us to conduct experiments testing the efficacy of tailoring to modality and view. The next steps are to conduct a series of experiments comparing the effect of filtering versus highlighting on cognitive load, while asking questions about participants' awareness of steps currently out of view (i.e., situational awareness [8]). We also plan to conduct studies using simple user models. These models will support compound filters such as filtering by several objects, e.g. truck 1 and airplane 1; filtering by other actors, e.g. what the other truck driver is responsible for; negation, e.g. do not show ship 1. We also plan to test different methods of filtering in plans with competing branches. Our research agenda includes a continued collaboration with industrial partners in the hydrocarbon exploration and unmanned vehicle domains.

References

1. Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. In: Symposium on Visual Languages. (1996) 336–343
2. Conati, C., Steichen, B., Tory, M., Buono, P., eds.: Workshop on User-Adaptive Visualization (WUAV) in association with UMAP. (2013)
3. Monroe, M., Lan, R., Plaisant, C., Shneiderman, B.: Temporal event sequence simplification. IEEE Trans. Visualization and Computer Graphics **19** (2013) 2227–36
4. Mellish, C., Evans, R.: Natural language generation from plans. Computational Linguistics **15**(4) (December 1989) 233–249
5. Henry, T.R.: Interactive graph layout: The exploration of large graphs. PhD thesis, The University of Arizona (1992)
6. Sarkar, M., Brown, M.H.: Graphical fisheye views of graphs. In: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM (1992) 83–91
7. Tintarev, N., Kutlak, R.: Sassy - making decisions transparent with argumentation and natural language generation. In: Workshop on interacting with Smart Objects in association with Intelligent User Interfaces. (2014)
8. Endsley, M., Jones, D.: Designing for Situation Awareness. CRC Press (2012)