An Intelligent User Interface for Mixed-Initiative Multi-Source Travel Planning

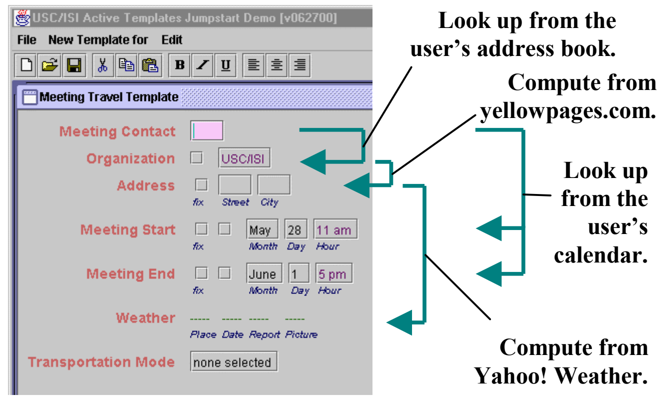
Martin Frank, Maria Muslea, Jean Oh, Steve Minton, Craig Knoblock
Information Sciences Institute, University of Southern California
4676 Admiralty Way; Marina del Rey, California 90292, USA
{frank, mariam, jeanoh, minton, knoblock}@isi.edu

ABSTRACT

A *mixed-initiative planner* in our context is one in which either the human or the computer can spontaneously provide the content of the same input fields. A *multi-source planner* is one that accesses multiple external information sources in parallel, using separate threads. This type of highly dynamic user interface is desirable but presents a challenge in "keeping the user in control" because it can be confusing to understand which fields of the form currently "belong" to the user, which ones "belong" to the system, how these two interact, and when and how their ownership changes.

INTRODUCTION

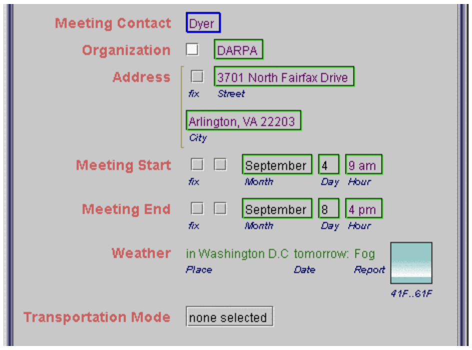
At present, using the Web for travel planning is cumbersome because there are several sites involved (one for air reservations, one for the hotel, another one for car rentals, all requiring re-entry of data). We developed a unified user interface in which a single (albeit complex and highly interactive) form can be used for planning all aspects of a trip. There are three key components involved. First, a multitude of external on-line sources must be ontologically modeled and be "wrapped" to access their data at run-time. This is similar in spirit to the light-weight model used by the Agent Playground presented at an earlier IUI conference [2] but uses the more sophisticated ontological modeling of Ariadne [3]. Second, a new breed of multi-threaded and highly interruptible constraint propagation system is needed to manage the flow of mixed-initiative field updates through the form [1]. Third, the human interface has to make the power of the system available to users but still keep them in overall control. The latter is the focus of this paper. The flow of data in an example travel application is depicted below.



In principle, all fields but the first (defining whom to meet) can be automatically defaulted by the system (presuming the person to meet is in the user's address book, the meeting dates are in the user's calendar, and the external Web sites are up). However, the user can also step in at any time and supply fields by hand (because the computation takes too long, produces less than ideal results, or because it fails outright).

Presume the user just chose the name of the person to meet (the first field above) from a menu of existing contacts. The screenshot below shows the system computing immediately after that event. The field that was entered by the user appears in blue, while the fields currently being computed are red. (There is also the option of marking the transitive closure of possibly affected fields in orange, which would show all remaining fields other than the Transportation Mode in that color.)

The associated computation stops if the user selects any field currently being computed with the mouse. In the example above, if the user selected May, all six fields in Meeting Start/End would have their red borders removed. If the user provides a new value for any field, this will retrigger the computation(s) that depend on it. The snapshot below shows the stable end state presuming that the user did not interfere with the system state depicted above.



Blue
Green
Green

Green

Green

Green

Green
(system-provided)

Now presume that the user realizes that while his contact works in Washington D.C., the meeting is actually going to occur in San Diego. In the screenshot below, she has changed the second field to a San Diego location ("SPAWAR-SD"). It turns blue (user-provided), and in addition it is now checked as "fixed", meaning that the computer will never overwrite its value in the future unless the user explicitly removes that check mark.

Blue
Blue
Green

Green

Green

Green

Blue
(user-provided)

If the user does un-check the checkbox, the system re-takes ownership of the field (it will turn green again) and re-triggers its computation. Thus, un-checking the box in the screenshot above will (eventually) re-create the state of the figure preceding it.

The screenshot above also shows our use of sub-sections to manage the complexity of the forms. They were expanded because the user explicitly chose to drive out and fly back. We now never automatically expand sub-forms even if the system can make a confident choice for the triggering field because it is visually too confusing for users.

LIMITATIONS

There are still several shortcomings in our current implementation: (a) The current interface does not use screen space very efficiently and – worse – replicates some fields many times in its sub-forms (that's good for understanding the local flow of control in the sub-form but overall makes the form longer than necessary – a flight trip with several legs and car rentals can result in a form that is three or more pages long). (b) The flow dependencies between fields are not obvious because they are not explicitly depicted in the interface (we considered simply drawing arrows between them but some dependencies are on far-away fields). (c) There can be a hierarchy of checkboxes for fixing fields (in the first figure of this paper, you can fix the entire date of the Meeting Start with the first checkbox, or just its month with the second) which was confusing. We plan to either draw appropriate enclosing rectangles for the checkmarks or simply offer only a single level of fixing.

CONCLUSION

This interface design (1) clearly marks the (human or system) origin of the fields' content, and (2) lets them "fix" information snippets so that they can make steady progress on the form and are assured the system will not override content that they provided. It is too early to claim that this technique suffices to overcome the inherent interface challenges of a mixed-initiative form, especially as we have not evaluated it outside our organization. However, we believe that the above limitations we ran into as well as the key lessons we learned as our mixed-initiative design evolved over the course of a year – (1) addressing confusion about the origin of field content and (2) assuring the users that they are in control and making progress -- may be of value to future designers of similar mixed-initiative forms.

ACKNOWLEDGEMENTS

We gratefully acknowledge Shalom Flank's contributions to this design. The research reported here was supported in part by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under contract number F30602-99-1-0524, in part by the Rome Laboratory of the Air Force Systems Command and DARPA under contract number F30602-98-2-0109. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policy or endorsements, either expressed or implied, of any of the above organizations or any person connected with them.

REFERENCES

1. Barish, G., DiPasquo, D., Knoblock, C., and Minton, S. A dataflow approach to agent-based information management. *International Conference on Artificial Intelligence (IC-AI 2000)*, Las Vegas, NV, 2000.
2. Frank, M., Szekely, P. Collapsible User Interfaces for Information Retrieval Agents. *IUI'99*, (Redondo Beach, CA, January 5-8) 1999, pages 15-22.
3. Knoblock, C. Minton, S., Ambite, J.-L., Ashish, N., Muslea, I. And Philpot, A., Tejada, S. The ARIADNE Approach to Web-based Information Integration *To appear, International Journal on Cooperative Information Systems (IJCIS)*.