Mixed-Initiative Workflow Composition

Marc Spraragen

University of Southern California / Information Sciences Institute
4676 Admiralty Way, Marina del Rey, CA 90292 USA +1 310 448 9197
marcs@isi.edu

Complex applications in many areas, including scientific computations and business-related systems, are represented as computational workflows composed out of multiple components. There are several approaches that help a user compose these workflows. Some composition systems implement a user-system interactive approach (Kim, Spraragen, and Gil 2004). These systems are useful for expressing user preferences during composition; however, they can be tedious to use if a large number of tasks are needed in the workflow, as composition is done one step at a time, manually. Another approach to workflow composition is full automation, which aims to eliminate unnecessary user interaction during composition (Blythe et al. 2003), (Myers et al. 2003). This approach is efficient, but is not ideal if user preferences need to be expressed during composition.

Our approach combines the strengths of manual and automatic approaches into mixed-initiative workflow composition. This combined approach uses automated planning techniques, while also incorporating user preferences during composition. Our approach is implemented in a new system, AutoCAT, by combining an interactive workflow editor (Composition Analysis Tool or CAT) (Kim, Spraragen, and Gil 2004) and a planner, Prodigy (Veloso et al. 1995).

CAT enables users to manually compose workflows by adding and removing tasks, and by linking the tasks together in order to supply data (i.e., linking task outputs) to the workflow's user-defined end results. Using AI planning principles, CAT checks the workflow for possible composition errors introduced by a user. For instance, the workflow's links are checked for correctness (e.g., "flight number" task output linked to "driver license number" task input is incorrect), and checks for unnecessary workflow tasks (for example, if the workflow's only end result is "flight reservation", then a "real estate service" task is unnecessary). CAT's approach also involves searching a knowledge base of task types in order to suggest to the user the next composition steps that should be taken (for example, "Suggestion: remove the real estate service task"). CAT also lets the user choose one out of several similar tasks (e.g., "do you want to add Travelocity flight reservation task, or Orbitz?") to add to the workflow. Lastly, CAT's knowledge base contains abstract task types, for example "abstract flight reservation task," which can be included in a workflow as placeholders. CAT will then suggest that users specialize abstract tasks; one such suggestion might be "specialize abstract flight reservation task to Travelocity or Orbitz."

AutoCAT extends CAT by incorporating Prodigy's automatic workflow completion. AutoCAT takes a partial workflow created by the user, and sends it to Prodigy. The representation of the workflow in Prodigy defines the problem's initial state as every task output from a workflow (i.e., all effects), and the goal state as every task input from the workflow (i.e., all preconditions). Prodigy processes the problem, while referencing AutoCAT's knowledge base for the available task types that can be added to the workflow. Prodigy then returns the shortest sequence of operators (i.e., tasks), based on the initial state, which supplies all necessary effects to achieve the goal state (if such a sequence exists). During planning, if Prodigy reaches a state where more than one similar operator can be applied, it will return the most specific common parent of these operators, as specified in the knowledge base's task ontology (e.g., Prodigy will choose "abstract flight reservation service" instead of choosing between Travelocity or Orbitz). AutoCAT incorporates the tasks from Prodigy into the workflow, and then presents the completed workflow to the user in case he/she wants to make further changes.

AutoCAT provides mixed-initiative workflow completion and correction for users, by the following interaction (available at any point during composition).

1) The user composes a partial workflow manually.
2) Algorithms in AutoCAT automatically remove incorrect links and unnecessary tasks from the workflow.
3) Prodigy is invoked to complete the workflow, and returns a sequence of tasks to AutoCAT (including abstract tasks).

4) AutoCAT formulates suggestion choices for specializing the abstract tasks.
5) AutoCAT returns the completed workflow, and suggestions, to the user.
6) The user applies suggestions or further composes the workflow manually.

The work outlined above demonstrates how mixed-initiative composition can help users compose workflows. An existing system, PASSAT, has been successful in a similar mixed-initiative approach using hierarchical task network (HTN) planning models (Myers et al. 2003), while AutoCAT uses operator-based planning models instead of HTN. Our future work will include user tests of AutoCAT. We will compare these results with results from previous and future user evaluations of CAT, in part to analyze AutoCAT's efficiency and user-friendliness with respect to CAT.

References

Blythe, J., et al. 2003. The Role of Planning in Grid Computing. In *Proceedings of the Thirteenth International Conference on Automated Planning and Scheduling*, 153-163. Menlo Park, Calif.: AAAI Press.

Kim, J., Spraragen, M., and Gil, Y. 2004. An intelligent assistant for interactive workflow composition. In *Proceedings of the Ninth International Conference on Intelligent User Interfaces,* 125-131. New York, New York: ACM Press.

Myers, K., et al. 2003. A Mixed-initiative Framework for Robust Plan Sketching. In *Proceedings of the Thirteenth International Conference on Automated Planning and Scheduling*, 256-266. Menlo Park, Calif.: AAAI Press.

Veloso, M., et al. 1995. Integrating Planning and Learning: The PRODIGY Architecture. *Journal of Theoretical and Experimental Artificial Intelligence,* 7(1):81-120.