Customizing the Composition of Actions, Programs, and Web Services with User Preferences[*]

Shirin Sohrabi

Department of Computer Science, University of Toronto
shirin@cs.toronto.edu

Abstract. Web service composition (WSC) – loosely, the composition of web-accessible software systems – requires a computer program to automatically select, integrate, and invoke multiple web services in order to achieve a user-defined objective. It is an example of the more general task of composing business processes or component-based software. Our doctoral research endeavours to make fundamental contributions to the knowledge representation and reasoning principles underlying the task of WSC, with a particular focus on the customization of compositions with respect to individual preferences. The setting for our work is the semantic web, where the properties and functioning of services and data are described in a computer-interpretable form. In this setting we conceive of WSC as an Artificial Intelligence planning task. This enables us to bring to bear many of the theoretical and computational advances in reasoning about action and planning to the task of WSC. However, WSC goes far beyond the reaches of classical planning, presenting a number of interesting challenges that are relevant not only to WSC but to a large body of problems related to the composition of actions, programs, business processes, and services. In what follows we identify a set of challenges facing our doctoral research and report on our progress to date in addressing these challenges.

1 Challenges and Background

Given a set of suitably described services, a specification of the state of the world, and a user objective, Web service composition (WSC) is the task of composing a set of services to achieve the user's objective. A popular approach to WSC is to characterize it as an Artificial Intelligence (AI) planning problem and to solve it as such (e.g., [8, 9, 1]). However, WSC goes far beyond the reaches of classical planning, presenting a number of interesting challenges that are relevant not only to WSC but to a large body of problems related to the composition of actions, programs, business processes, and services. For example, unlike typical classical planning tasks, knowledge of how to achieve the user's objective is often known, at least at an abstract level; plans are often plentiful, high-quality plans are not, requiring optimization of complex preferences. Another differentiating property

[*] The doctoral research described in this paper is being developed and carried out under the supervision of Professor Sheila McIlraith.

of WSC is that it can be data intensive resulting in planning domains with tens of thousands of actions, each of which is itself a program with non-determinism and intermediate state.

Several planning-based approaches have appealed to some sort of a template or workflow structure to help define the user objective and/or to guide the composition. The agent programming language Golog (e.g., [9]), Hierarchical Task Networks (HTNs) (e.g., [14]), and Finite State Automata (FSA) (e.g., [3]) have all been used for this purpose. In the case of our work, our objective is to define a **flexible, declarative WSC template** that provides high-level guidance on how to perform a task, but can leave many of the details to run-time synthesis, should that be warranted. For many WSC problems, the task can be realized by a diversity of different services, offering comparable, but not identical services. Also unknown at the outset is the data that serves as choice points in a WSC – the availability of goods, their properties and pricing, etc. A flexible composition template streamlines the generation of a composition, while enabling the individual user to customize the composition with respect to their preferences and constraints perhaps in association with preferences and constraints of other stakeholders such as the corporation they work for, the laws of the countries in which they are doing business, etc.

The general challenge we face in our doctoral research is to investigate principled techniques for composing web services, that support user customization. This manifests itself in a number of specific research challenges. We discuss some of the major challenges below.

Challenge 1: The first challenge is how to customize WSC templates (the problem specification) to meet the objectives of individual users. The templates are meant to be flexible yet shared by many users. So the challenge is how to customize the templates for each individual user. In order to do so we need to address how users can specify their preferences and to that end propose a rich preference language with which to express arbitrarily complex and mutually inconsistent preferences. In addition, we would like to have a user preference language that can handle preference specifications over both the functional and the non-functional properties of services such as trust, reliability, and privacy.

Challenge 2: Our second challenge is how to synthesize compositions that adhere to policies and regulations. Policies and regulations are an important aspect of semantic web services. Many customers are concerned with enforcement of regulations, perhaps in the form of corporate policies and/or government regulations. Software that is developed for use by a particular corporation or jurisdiction will have the enforcement of such regulations built in. For web services that are published for use by the masses this is not the case, and the onus is often on the customer to ensure that regulations are enforced when a workflow is constructed from multiple service providers. For inter-jurisdictional or international business, different regulations may apply to different aspects of the composition. Hence, providing a mechanism for generating compositions from templates that adhere to such regulations is the second challenge we would like to address.

Challenge 3: Our third challenge is how to develop heuristics to search efficiently for an optimal plan/composition. An optimal plan is a plan that has the best quality (i.e., no other plan has a better quality than this plan). Heuristic-guided search is an effective method for efficient plan generation, but the challenge is to find a heuristic that gives guidance towards optimal solutions without exhaustively searching the search space. We can use either an admissible or inadmissible set of heuristics. An admissible heuristic is a heuristic that never overestimates the cost of reaching the goal. If an admissible heuristic is used in a A*-like algorithm, then first plan found would be an optimal plan. However, finding such plan in practice may not be feasible when the search space is large. Hence, we may consider using inadmissible heuristics with a hope of finding a good-quality plan quickly instead and find a condition under which we can guarantee optimality despite the use of inadmissible heuristics.

Challenge 4: Our fourth challenge is how to perform and integrate online information gathering in order to gather the necessary information needed to produce high-quality compositions in the absence of complete information. Many planning-based characterizations of WSC make an assumption that there is complete information about the initial state. This assumption is often violated in many real-world settings; it is impractical or impossible to have all the information necessary to generate a composition prior to the commencement of the search for a composition. A more compelling solution is to instead gather information as it becomes necessary in the generation and optimization of the composition. In our doctoral research, we describe a means of executing web services to collect information as it is deemed necessary to inform the search for a valid, ideally optimal, composition.

Challenge 5: Our fifth challenge is how to optimize compositions with tens of thousands of services and with extensive data. Optimization requires considering all alternatives, at least implicitly. However, given the large volume of information available on the Web, evaluating the search space effectively is a challenging problem that has not been addressed in previous work. Consider just as an example three information sources A, B, and C containing i, j, and k data items, respectively. In the worst case, the system will need to access and evaluate $i*j*k$ different alternatives in order to identify the optimal combination. However, if the choice of data for C can be made independently of A and B, then the search space is worst case $k + i * j$. This decomposition and *localization* of the optimization is one of the ways that humans manage to address the complexity of optimization tasks, and it's one that we plan to incorporate into our work.

In the next section we will briefly overview a small subset of the previous work and then we will describe the progress we have made so far in addressing the above five challenges. We develop knowledge representation and reasoning techniques, describe formal properties, prove properties of our formalizations including, soundness, completeness, correctness and optimality, where relevant. We also evaluate our work experimentally to demonstrate the practical advantages of our approach.

2 Related Work

There is an important body of previous work that our work builds on. We note a subset of them here. In particular, a number of researchers have advocated using AI planning techniques to address the task of web service composition including planners that are based on model checking (e.g., [21]) and planners that use a regression-based approach [8]. Previous work has also considered using a template or workflow to ease the task of composition including the work using Golog [10, 9] and HTNs [14, 13, 7]. The work on the so-called Roman model is another example of a template-like approach (e.g., [3]). Also there are several proposed solutions to the information gathering problem (e.g., [9, 14, 6]); however, they have not examined the problem of information gathering in the context of optimizing the composition. In the absence of the need to optimize, it is often sufficient to arbitrarily select one choice among many and to ignore the rest. However, if the task is to generate a high-quality composition that optimizes for the user's preferences, then the entire space of alternatives must be considered somehow. This both alters the information gathering task and also greatly increases the search space for a composition.

While no other WSC planners can perform true preference-based planning, SHOP2 [11] and ENQUIRER [6] handle some simple user constraints. Also a notable work is the SCUP prototype planner in [7] but there are several differences to our work. In particular, their preferences are pre-processed into task networks and conflicting user preferences are detected and removed prior to invocation of their planner. Further, they do not consider handling regulations and are not able to specify preferences over the quality of services.

3 Progress Made To Date

Progress towards addressing the challenges presented in my doctoral research have appeared in a number of publications as summarized in Figure 1.

As mentioned above, a composition template can be represented in a variety of different ways. One way to represent a template is to use a workflow or a flowchart. This can be expressed pictorially as a schematic or alternatively in a form akin to a procedural programming language. The Algol-inspired Golog agent programming language provides one such procedural language (e.g., [12]). Indeed, the first template-based approach to WSC exploited Golog to provide a so-called generic procedure that provided a template specification of the composition [10, 9]. The Golog procedures were combined with individual user constraints (e.g., *"I want to fly with a star alliance carrier"*) at run time, resulting in *dynamic binding of web services*. However, the user constraints considered were hard constraints, i.e., realizations that did not satisfy those constraints were eliminated. In [20], we make progress towards addressing our first challenge by extending this framework to deal with *soft* user constraints (i.e., preferences). To specify user preferences, we exploit a rich qualitative preference language, proposed by Bienvenu et al. to specify users' preferences in a variant of linear

#	**Challenge**	**Approach**	**Progress**
1	Customize WSC templates for each user	Use and extend preference languages LPP and PDDL3	[20, 17] [16, 18]
2	Generate compositions from templates that adhere to regulations or policies	Use pruning and specify regulations as LTLs	[18]
3	Develop heuristics to search efficiently for an optimal composition	Use (in)admissible heuristic techniques developed for planning	[15, 16]
4	Access & integrate online information gathering to produce a high-quality plan	Use a middle-ground execution engine	[19]
5	Optimize compositions with many services and with extensive data	Use preference decoupling	[19]

Fig. 1. Challenges in our doctoral research and the progress made to address them.

temporal logic (LTL) called LPP [2]. We prove the soundness of our approach and the optimality of our compositions with respect to the user's preferences. Our system can be used to select the optimal solution from among families of solutions that achieve the user's stated objective. Our system, GologPref, was implemented in Prolog and integrated with a selection of scraped web services that are appropriate to our test domain of travel planning. Unfortunately, the implementation of the system was not optimized.

Similar to Golog, HTNs [5] provide useful control knowledge — advice on how to perform a composition. However, this how-to knowledge is specified as a *task network*. The task network provides a way of hierarchically abstracting the composition into a set of tasks that need to be performed and that decompose in various ways into leaf nodes realized by programs. While HTNs specify a family of satisfactory plans, they are, for the most part, unable to distinguish between successful plans of differing quality. In [17] we address the problem of generating preferred plans by combining the procedural control knowledge specified by HTNs with rich qualitative user preferences. Note this is work towards addressing our first challenge where templates are now specified in HTNs. The outcome of our work is a language for specifying user preferences, tailored to HTN planning, together with a provably optimal preference-based planner, **HTNPlan**, that is implemented as an extension of SHOP2 [14], a highly-optimized HTN planner for the task of WSC. To specify user preferences, we augment the preference language LPP used in [20] with HTN-specific constructs. Among the HTN-specific properties that we add to our language, is the ability to express preferences over how tasks in our HTN are decomposed into subtasks, preferences over the parameterizations of decomposed tasks, and a variety of temporal and non-temporal preferences over the task networks themselves. To compute preferred plans, we propose an approach based on forward-chaining heuristic search. Our heuristic uses an admissible evaluation function measuring the satisfaction of preferences over partial plans. We prove our approach sound and optimal with respect to the plans it generates by appealing to a situation calculus [12] semantics of our preference language and of HTN planning.

HTNPlan discussed briefly above is a provably optimal preference-based planner; however, with large search space, finding this optimal plan may not be feasible. As an alternative, in [16] we propose several *inadmissible* heuristics, designed specifically to guide the search quickly to a good decomposition. In particular, we designed a heuristic called the *look-ahead* heuristic that is designed specifically to address this problem. Also we decided to use the popular Plan Domain Description Language, PDDL3 [4] as our preference language instead. Similar to [17] we extended PDDL3 to support specification of preferences over HTN constructs; note this is work towards addressing our first and third challenge. To compute preferred plans, we propose a branch-and-bound algorithm, together with our set of heuristics that leverage HTN structure. The search is performed in a series of episodes, each of which returns a plan with a better quality than the last plan returned. We showed that under some condition we can guarantee optimality. The experimental evaluations of our planner shows that our HTN preference-based planner, **HTNPlan-P**, generates plans that, in all but a few cases, equal or exceed the best preference-based planners in plan quality. As such, our results shows that our approach is viable and promising to preference-based planning.

We tackle our first and second challenge in [18] by providing a mechanism for generating compositions from templates that adhere to regulations as well as extending the preference language developed earlier with preferences over service and data selection. We specify regulations as a subset of Linear Temporal Logics (LTLs), considering for the most part the *never* and *always* constructs, and use pruning to eliminate those compositions that violate such regulations. Hence, we describe a composition framework that that simultaneously optimizes, at run time, the selection of services based on functional and non-functional properties and their groundings, while enforcing stated regulations. We also use the search heuristics developed in our previous work and provide an implementation that combines HTN templates, the optimization of rich user preferences, and adherence to LTL regulations all within one system. Experimental evaluation on our system, **HTNWSC-P**, shows that our approach can be scaled as we increase the number of preferences and the number of services.

4 Current and Future Research

Much of the AI-related work on WSC that relates it to an AI planning problem performs composition offline prior to execution. Recent research on WSC has argued convincingly for the importance of optimizing quality of service and user preferences. While some of this optimization can be done offline, many interesting and useful optimizations are data-dependent, and must be done following execution of at least some information-providing services. In our recent work [19], we examine this class of WSC problems, attempting to bridge the gap between offline composition and online information gathering with a view to producing high-quality compositions without excessive data gathering. Our investigation is performed in the context of our preference-based HTN web service composition system [18]. We propose a way to address the critical information-gathering component of preference-based WSC as well as optimization. This need to actually execute services to gather data, as well as the potential size and nature of the resultant optimization problem truly distinguishes our WSC task from previous work on preference-based planning. To this end we propose a notion of middle-ground execution that enables information gathering during generation of a WSC. We further propose a notion of *localized data optimization* in which the optimization task can be decomposed into smaller, local optimization problems, while preserving global optimality. We showed that our approach to data optimization can greatly improve both the quality of compositions and the speed with which they are generated.

In future work, we plan to further improve our approach, ideally in addressing all of our challenges. In particular, in addressing our second challenge, we specified regulations in a subset of LTLs and we plan to further improve our work by considering the full expressive power of LTLs instead. Furthermore, we plan to improve our online information gathering procedure. Currently, the HTN structure embodies the place where information gathering is necessary. In future, we like to improve our procedure to remove this restriction by possibly having a pre-processing step influenced by query optimization techniques to remove or push forward the “hard” information gathering steps. Finally, we would like to design a user-friendly interface that possibly not only takes the user’s preferences, objectives, policies, but also interacts with the user in a mixed-initiative manner during the composition construction time.

In conclusion, the need to compose actions, programs and web service is pervasive not only on the Web but in general software engineering or manufacturing settings where we like to describe and possibly reuse software with the desire to customize. We hope to make principal theoretical and practical contributions to these fundamental problems.

Acknowledgements

We gratefully acknowledge funding from the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Ontario Ministry of Innovations Early Researcher Award (ERA). Note some of the materials presented here has appeared in parts in previous publications as noted in Figure 1.

References

1. Bertoli, P., Kazhamiakin, R., Paolucci, M., Pistore, M., Raik, H., Wagner, M.: Continuous orchestration of Web services via planning. In: Proc. of the 19th Int’l Conference on Automated Planning and Scheduling (ICAPS). pp. 18–25 (2009)
2. Bienvenu, M., Fritz, C., McIlraith, S.: Planning with qualitative temporal preferences. In: Proc. of the 10th Int’l Conference on Knowledge Representation and Reasoning (KR). pp. 134–144 (2006)

3. Calvanese, D., Giacomo, G.D., Lenzerini, M., Mecella, M., Patrizi, F.: Automatic service composition and synthesis: the Roman Model. IEEE Data Eng. Bull. 31(3), 18–22 (2008)
4. Gerevini, A., Long, D.: Plan constraints and preferences for PDDL3. Tech. Rep. 2005-08-07, Department of Electronics for Automation, University of Brescia, Brescia, Italy (2005)
5. Ghallab, M., Nau, D., Traverso, P.: Hierarchical Task Network Planning. Automated Planning: Theory and Practice. Morgan Kaufmann (2004)
6. Kuter, U., Sirin, E., Nau, D.S., Parsia, B., Hendler, J.A.: Information gathering during planning for Web service composition. In: Proc. of the 3rd Int'l Semantic Web Conference (ISWC). pp. 335–349 (2004)
7. Lin, N., Kuter, U., Sirin, E.: Web service composition with user preferences. In: Proc. of the 5th European Semantic Web Conference (ESWC). pp. 629–643 (2008)
8. McDermott, D.V.: Estimated-regression planning for interactions with Web services. In: Proc. of the 6th Int'l Conference on Artificial Intelligence Planning and Scheduling (AIPS). pp. 204–211 (2002)
9. McIlraith, S., Son, T.: Adapting Golog for composition of semantic Web services. In: Proc. of the 8th Int'l Conference on Knowledge Representation and Reasoning (KR). pp. 482–493 (2002)
10. McIlraith, S., Son, T., Zeng, H.: Semantic Web services. IEEE Intelligent Systems. Special Issue on the Semantic Web 16(2), 46–53 (March/April 2001)
11. Nau, D.S., Au, T.C., Ilghami, O., Kuter, U., Murdock, J.W., Wu, D., Yaman, F.: SHOP2: An HTN planning system. Journal of Artificial Intelligence Research 20, 379–404 (2003)
12. Reiter, R.: Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems. MIT Press (2001)
13. Sirin, E., Parsia, B., Hendler, J.: Template-based composition of semantic Web services. In: AAAI-05 Fall Symposium on Agents and the Semantic Web (2005)
14. Sirin, E., Parsia, B., Wu, D., Hendler, J., Nau, D.: HTN planning for Web service composition using SHOP2. Journal of Web Semantics 1(4), 377–396 (2005)
15. Sohrabi, S., Baier, J., McIlraith, S.A.: HTN planning with quantitative preferences via heuristic search. In: 8th International Conference on Automated Planning and Scheduling (ICAPS) Workshop on Oversubscribed Planning and Scheduling (2008)
16. Sohrabi, S., Baier, J.A., McIlraith, S.A.: HTN planning with preferences. In: Proc. of the 21st Int'l Joint Conference on Artificial Intelligence (IJCAI). pp. 1790–1797 (2009)
17. Sohrabi, S., McIlraith, S.A.: On planning with preferences in HTN. In: 12th International Workshop on Non-Monotonic Reasoning (NMR-08). pp. 241–248 (2008)
18. Sohrabi, S., McIlraith, S.A.: Optimizing Web service composition while enforcing regulations. In: Proc. of the 8th Int'l Semantic Web Conference (ISWC). pp. 601–617 (2009)
19. Sohrabi, S., McIlraith, S.A.: Preference-based Web service composition: A middle ground between execution and search. In: 9th International Semantic Web Conference (ISWC) (2010), to appear
20. Sohrabi, S., Prokoshyna, N., McIlraith, S.A.: Web service composition via generic procedures and customizing user preferences. In: Proc. of the 5th Int'l Semantic Web Conference (ISWC). pp. 597–611 (2006)
21. Traverso, P., Pistore, M.: Automatic composition of semantic Web services into executable processes. In: Proc. of the 3rd Int'l Semantic Web Conference (ISWC) (2004)