

# Web Services and Automated Planning for Intelligent Calendars

George Markou, Anastasios Alexiadis, Ioannis Refanidis

Department of Applied Informatics, University of Macedonia  
Thessaloniki, Greece

`gmarkou@uom.gr`, `tallex@java.uom.gr`, `yrefanid@uom.gr`

**Abstract.** This paper promotes the automated creation of hybrid personal plans, comprising web services and real human activities, to be supported by the next generation of intelligent calendar applications. A prototype work is present-ed, utilizing both atomic web services and composite ones, the latter having been previously generated from a contingent web service composition module, aiming at in-creasing the likelihood of achieving their intended goal. A metric planner generates a plan, giving priority to web service calls over human activities. Then, a scheduler schedules the human activities into the user’s calendar, taking into account the ordering constraints that result from the plan. The resulting schedules substitute human activities with web services, thus increasing the user’s capacity, his free time, as well as the scheduling options. As a proof of concept we present a case study implementation utilizing existing state-of-the-art components.

**Keywords:** web services \* intelligent calendar \* personal activities

## 1 Introduction

Although in the recent past the popularity of paper calendars has steadily declined in favor of web-based calendar applications like Google Calendar, such applications still do not provide any means for automated activity scheduling. In that way, users are forced to decide for themselves whether a particular activity has enough time to be scheduled alongside another, or whether the distance between the places where the two activities occur is a prohibiting factor.

This problem was efficiently tackled in our previous work, by SELFPLANNER [1], a web-based calendar application that combines a rich problem model with a fast domain-specific scheduler and automatically produces optimized schedules. SELFPLANNER also uses Google Maps to compute the necessary travelling times between the activities’ locations, as well as Google Calendar to present the outputted schedules.

The obvious next step in intelligent calendar applications is to employ planning, instead of pure scheduling, to achieve the user’s goals. Existing systems, like SELFPLANNER, rely on the user to select the activities to be included in his plan, with

the system providing only scheduling functionalities. We envision a situation where intelligent calendar applications (a) support action ontologies, with action descriptions containing pre-conditions and effects, (b) allow the user to set his goals, (c) know the user's state, and (d) employ auto-mated planning to create plans that achieve the user's goals.

An even further step would be to extend intelligent calendar applications so as to take web services into account. Web services can be considered as normal actions that can be used to achieve users' goals or other actions' preconditions. Incorporating web services into the action ontology enables the substitution of human activities by web service calls, thus allowing for more flexible plans, more goals to achieve or just more free time. As an example of such a setting that is very common in our daily routine consider this: a person may wish to attend a concert, and for that reason he may insert a personal activity in his schedule so as to reserve time and remember to buy tickets for it. To achieve this goal the user is required to physically go to a brick-and-mortar shop or buy his tickets online: both options re-quire some of his time (obviously, buying the tickets online requires less time). A more efficient electronic calendar, on the other hand, would have searched for an alternative - automated - way of achieving such goals, so as to relieve the user from the burden of manually executing the necessary actions.

In this work we propose such an approach; it is based on the integration of web services with an existing metric planner and an intelligent calendar application, namely SELFPLANNER. Web services may be simple or composite; in the latter case, a contingent web service composition module has been used, trying to reduce the non-determinism underlying their execution. From the planning perspective, though, web services are considered deterministic. As a result of the above process, the user's schedule contains only the human activities that cannot be completed through the use of web services, thus relieving him of the extra burden of achieving the rest.

The rest of the paper is structured as follows: First we present related work; next, we give a motivating example; then we present our approach and, finally, we conclude the paper and pose future directions of research.

## 2 Related Work

SELFPLANNER was the first system to tackle the problem of automated scheduling personal activities into electronic calendars, using a combination of greedy optimization algorithm, namely a modified version of the Squeaky Wheel Optimization (SWO) [2], and stochastic local search. SELFPLANNER employs a rich model supporting temporal domains and preferences, locations, interruptible and periodic activities, binary constraints and preferences, etc.

Bank et al. [3] build directly upon this work, using SWO in addition to a set of calendar entity types that they propose; in specific, they discriminate between simple events, multiple choice events, floating events and tasks. Moreover, they incorporate elements of psychology into the generation of the schedules, by defining preferences

such as that there should be no wasted travel time between events, or that creative tasks should be split into multiple segments.

La Placa, Pigot and Kabanza [4] follow a different approach, by utilizing Hierarchical Task Network (HTN) Planning [5] and focusing on a specific user target group. Their approach is directed towards people with cognitive impairments, e.g., Alzheimer's disease, and as such, HTN planning, which decomposes tasks into subtasks, is well suited as it resembles the way medical professional actually plan for their patients. Moreover, this degree of granularity is dependent on the specific patient, with information such as his impairment or personality being taken into account.

Finally, Berry et al. [6] present Emma, a personalized calendar management tool, which simultaneously manages calendars from multiple sources, with the main aims of facilitating the coordination of groups of people, the negotiation of their meeting times and the (re)scheduling of various events.

In regard to non-deterministic web service composition, Kuzu and Cicekli [7] present a conversion schema from OWL-S to PDDL and utilize an existing PDDL planner, namely Simplanner [8], to tackle non-determinism, through interleaving planning and execution. Zou et al. [9] follow a similar approach, albeit to generate a distributed plan; first, a web service choreography problem that contains explicit user defined contingencies is translated into a deterministic planning one and then, either FF [10] or SatPlan06 [11] are employed to solve it.

Dacosta et al. [12] on the other hand, opt for a stratified method so as to produce robust plans, which allow for semantic web services that achieve the same tasks. The approach generates a graph that contains all the possible contingency plans, with each path in it being a possible execution path, and each child node comprising an alternative execution possibility. Redundant operations have been removed from the graph, and the set of paths is ordered from the best – the one containing the smaller number of web services – to the worst.

In our previous work [13] we implemented MAPPPA, a cost sensitive probabilistic contingent planning approach specifically targeted for automated semantic web service composition. MAPPPA produces a contingent plan by integrating alternative deterministic plans previously computed in an anytime fashion from a determinized version of the original problem. It does so, however, without disregarding the information that each web service contains in relation to its execution cost and probability of alternative outcomes, thus, generating considerably more informed plans than other determinization approaches.

### 3 Motivating Example

Let us imagine a usual week of a Bob, who uses a web-based calendar application to organize his time. Due to being self-employed, Bob needs to spend all working hours at his office, to which he commutes every day from his home. Once every week, he watches a movie, and although he prefers to go to the cinema, sometimes he watches the movie at home, depending on his work schedule. However, if the movie's duration is such that it will end later than 11 pm, he does not desire to watch a movie at

all, as he has to sleep early. Moreover, this week, he will travel on a business trip abroad; a day before the trip, the day he usually watches a movie, he has to book his airplane tickets and hotel, as well as to buy a travel guide for the city he will visit.

In order to schedule these activities and insert them into a calendar, the user has to define their temporal domain and, if they are interruptible, the minimum and maximum allowed duration for their parts. Moreover, for each activity, the user has to declare whether it is periodic or not, as well as if it is bound to specific locations. For example, the user should define his daily work as a periodic task, with a temporal domain from 8 a.m. to 5 p.m., and a minimum and maximum duration of its parts – as he cannot work continuously, e.g., 30 minutes and 2 hours respectively. Watching a movie is also periodic but non-interruptible, and has a temporal domain set late in the evening, with a minimum and maximum duration of 90 minutes and 180 minutes respectively.

Since the user's work requires his physical presence, only a human activity can be placed in his calendar. However, for the rest of the aforementioned tasks, a combination of human and web services' activities can be inserted. The user may have to drive to the cinema, choose among the available movies there and buy the tickets himself. Alternatively, he may ask for a list of the available movies to be emailed to him, book the tickets online and then, having saved considerable time, travel to the cinema later. Another option altogether would be to rent a movie online and watch it at home. As to the user's business trip, again, there are various alternative activities; the user may visit a single tourist agent to book his tickets and hotel; or prefer an online reseller. He may also require another trip to a bookstore to purchase his travelling guide or purchase it online and have it sent at home.

Since these options create a complex problem, containing a multitude of constraints and preferences, a schedule manually created by a human is usually highly inefficient. Applications such as SELFPLANNER tackle this problem; however, they only deal with human activities and, as such, they cannot take advantage of the opportunities that are offered by the use of web services. In the aforementioned scenario, since a human activity requires the user to first purchase a movie ticket himself, in certain situations he may not have had the time to do so, and he would have had to watch it at home instead. Even worse, if he had to visit a travel agent and a bookstore, he may not have even had the time to watch a movie at home.

With the introduction of web services, the user saves the time needed to perform the actual action of purchasing these services and to travel between locations.

## **4 Proposed Approach**

This work assumes that web services are semantically annotated and that their descriptions are present in an online registry; in previous work, we presented such a registry [14], which, furthermore, provides a translation of each web service to a PDDL action. Moreover, this registry also contains composite web services fortified against non-determinism, having been generated by MAPPPA prior to the start of the scheduling process. Thus, the composite web services used comprise multiple execu-

tion paths achieving the same result, and for this reason can only fail when all the execution paths fail.

Fig. 1 presents the proposed system’s architecture. Initially, we employ an existing metric planner, namely LPG-td [15]; this step is necessary in order to automate the planning process as the activities in SELFPLANNER are normally entered by the users. Instead, in this case, in order to obtain the set of activities that achieve the desired goal a planner has to be utilized. LPG-td receives as input a planning problem containing both web services and human activities.

This problem comprises of the translation of the web services from the registry, as well as a simplified version of the human activities; that is, the locations, durations and temporal domains of the activities, along with any preferences and constraints in regard to them are removed. The metric planner does not differentiate between the two types of activities; as a result, it also treats all web services as deterministic ones, i.e., as if their intended output (the most probable one) is always outputted. Moreover, the planner has to take into account that web services are preferred to their human activities’ counterpart. This is achieved by setting the cost of human activities higher than that of web services.

LPG-td is capable of generating a sequence of alternative plans, each being an improvement – in terms of the specified metric – compared to the previous one; the plans generated by LPG-td allow for parallel actions and, thus, are very similar to partial order ones.

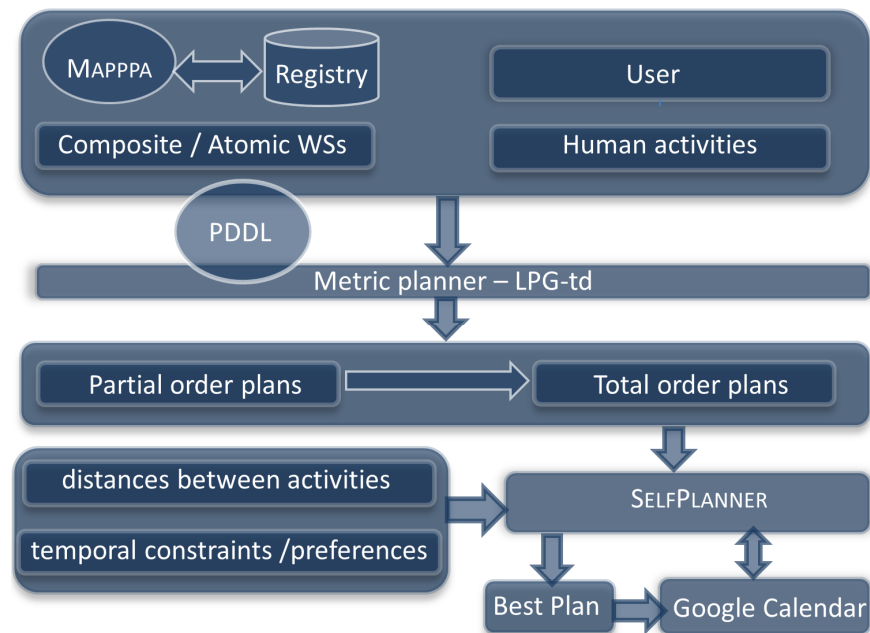


Fig. 1. Proposed system’s architecture overview.

We feed the best generated plan by LPG-td to SELFPLANNER so as to create a detailed schedule. In case a feasible plan does not exist, the rest of the previously generated alternative partial order plans are attempted to be scheduled in a similar process. In order to schedule the activities of the partial order plan, SELFPLANNER employs the information concerning temporal constraints and preferences (loaded from a separate activity definition file for a given problem instance), as well as the actual distances between the activities' locations as returned by the Google Maps Distance Matrix API.<sup>1</sup> Web services are considered to have an open temporal domain and are not related to a specific location, as they can be executed at any time and place. Moreover, they can be scheduled in parallel to human activities. SELFPLANNER can generate multiple alternative plans; the human activities of the best schedule are uploaded into the user's Google calendar.

## 5 Conclusions and Future Work

This paper presents the first steps towards the next generation of intelligent calendar applications. We propose an approach comprising a contingent web-service composition system, a partial order metric planner and a scheduler to insert human activities into a user's web calendar. The contingent planner is used to create composite web services, able to achieve complex goals with high probability of success; the metric planner is used to select an optimized set of activities to achieve a user's goal, favoring web service calls than real activities; and, finally, the scheduler automatically produces an optimized plan based on the user's constraints and preferences.

The paper also presents a motivating example along with a – still under work – implementation. We aim to further improve this implementation, first by providing a graphical interface, as well as by integrating it with a web service execution platform. Also, we propose to perform an exploratory evaluation, by providing the - users with two schedules, one consisting solely of human activities and an equivalent one comprising both human and web service activities, and having them rate each one online.

Finally, various research and implementation issues are still open; for example, we assume that the user's schedule should only contain the human activities that cannot be substituted by web services; this is achieved through setting the cost of human activities higher than that of web services. However, in some cases this may not be true; employing a web service may incur a (financial) cost that the user does not prefer over the benefit of not employing the respective human activity. Moreover, a web service activity may not always provide exactly the same functionality or satisfaction as its human counterpart; such a case is present in our motivating example, in which a user prefers to go to the cinema to watch a movie, than watching it at home through streaming. In this case, however, if the problem is modeled so as to favor the human activity, it could be impossible to include it in the plan, thus providing the use of a web service as an alternative plan. Such problems require further investigation.

---

<sup>1</sup> <https://developers.google.com/maps/documentation/distancematrix/>

**Acknowledgment.** This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF) - Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.

## References

1. Refanidis, I., Alexiadis, A.: Deployment and evaluation of SELFPLANNER, an automated individual task management system. *Comput. Intell.* 27(1), 41-59 (2011)
2. Joslin, D., Clements, D.: Squeaky wheel optimization. *J. Artif. Intell. Res.* 10, 353-373 (1999)
3. Bank, J., Cain, Z., Shoham, Y., Suen, C., Ariely, D.: Turning personal calendars into scheduling assistants. In: *Extended Abstracts of the 30th CHI Conference on Human Factors in Computing Systems* (2012)
4. La Placa, M., Pigot, H., Kabanza, F.: Assistive planning for people with cognitive impairments. In: *Proceedings of the Workshop on Intelligent Systems for Assisted Cognition hosted by the 21st International Joint Conference on Artificial Intelligence* (2009)
5. Ghallab, M., Nau, D., Traverso, P.: *Automated Planning: Theory and Practice*. Morgan Kaufmann (2004)
6. Berry, P. M., Donneau-Golencer, T., Duong, K., Gervasio, M., Peintner, B., Yorke-Smith, N.: Evaluating user-adaptive systems: Lessons from experiences with a personalized meeting scheduling assistant. In: *Proceedings of the 21st Innovative Applications of Artificial Intelligence Conference*, pp. 40–46 (2009)
7. Kuzu, M., Cicekli, N.: Dynamic planning approach to automated web service composition. *Appl. Artif. Intell.* 36(1), 1-28 (2012)
8. Onandia, E., Sapena, O., Sebastia, L., Marzal, E.: SimPlanner: An execution-monitoring system for replanning in dynamic worlds. In: *Proceedings of the 10th Portuguese Conference on Artificial Intelligence*, pp. 393-400 (2001)
9. Zou, G., Chen, Y., Xu, Y., Huang, R., Xiang, Y.: Towards automated choreographing of web services using planning. In: *Proceedings of the 26th AAAI Conference on Artificial Intelligence* (2012)
10. Hoffmann, J.: FF: The Fast-Forward Planning System. *AI Mag.* 22(3), 57-62 (2001)
11. Kautz, H., Selman, B., Hoffmann, J.: SatPlan: Planning as Satisfiability. In: *Proceedings of the 5th International Planning Competition* (2006)
12. Dacosta, L.A.G., Pires, P.F., Mattoso, M.: Automatic Composition of Web Services with Contingency Plans. In: *Proceedings of the 2nd International Conference on Web Services Workshop* (2004)
13. Markou, G., Refanidis, I.: Anytime planning for web service composition via alternative plan merging. In: *Proceedings of the 26th IEEE International Conference on Tools with Artificial Intelligence*, pp. 91-98 (2014)
14. Markou, G., Refanidis, I.: Composing semantic web services online and an evaluation framework. *Int. J. on Adv. in Internet Tech.* 6(3-4), 114-131 (2013)
15. Gerevini, A., Saetti, A., Serina, I., Toninelli, P.: Fast Planning in Domains with Derived Predicates: An Approach Based on Rule-Action Graphs and Local Search. In: *Proceedings of the 20th AAAI National Conference on Artificial Intelligence* (2005)