Temporal and Resource Controllability of Workflows Under Uncertainty

Matteo Zavatteri[1]

Department of Computer Science, University of Verona,
Strada le Grazie 15, 37134 Verona, Italy
matteo.zavatteri@univr.it

1 Context and Motivation

Workflow technology has emerged as one of the leading technologies for modeling, (re)designing and executing business processes in several different application domains such as industrial R&D, manufacturing, energy distribution, banking processes, critical infrastructures and healthcare. A workflow is the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules. The conceptual modeling of workflows underlying business processes has been receiving increasing attention over the last years and many technical aspects have been discussed, including flexibility, structured vs. unstructured modeling, change management, authorization models, temporal features, resource allocation, failure resistance and constraints (see, e.g., [14,15,18,19,21]).

Recently, attention was devoted in particular to the issue of expressing *temporal features* of workflows, such as task-duration constraints, temporal constraints between non-consecutive tasks, delays, deadlines and so on [14]. Properties of such temporal workflow models have been defined and analyzed. One of the most interesting is that of *dynamic controllability*, ensuring that a workflow can be executed satisfying all the given temporal constraints without the workflow management system restricting and/or controlling task durations but only assuming that each duration is within a specified range (*temporal uncertainty*) [14].

The authors of [14] also tackled dynamic controllability under another kind of uncertainty, *conditional uncertainty*, represented by the fact that some subsets of tasks have to be executed if and only if some conditions (abstracted as Boolean propositions) are true. Similarly to what happens for uncontrollable task durations, the truth-value assignment to such propositions is *out of control*. For instance, when a patient enters the emergency room, the severity of his condition is not known a priori but it is established by a physician, while the workflow is being executed. Since such a condition discriminates which tasks have, or have not, to be executed (i.e., the choice of the workflow path), the workflow management system must be able to get to the end of the workflow satisfying all relevant temporal constraints regardless of which tasks have to be executed and which task durations have to be satisfied. In [14], the authors did not consider workflow instances specifying both controllable and uncontrollable

workflow paths such that the choice of a controllable workflow path may exclude the choice of an uncontrollable one and vice versa. That is, they did not address *fallback temporal plans*, i.e., plans in which making a decision may exclude some uncontrollable part whose behavior risks violating some constraint.

Workflows also deal with the management of associated resources in order to complete business processes. In this thesis I considered, from a security point of view, the most trivial of resources: *users*. When we talk about security in the business process context, we must also talk about access control models, security policies and authorization constraints.

Therefore, an *access-controlled workflow* extends a classical workflow by adding users and authorization constraints. Users are authorized for tasks whereas authorization constraints say which users remain authorized for which tasks depending on who did what. *Role-based access control models (RBAC, [20])* put another layer of security on top of access controlled workflows injecting the concept of *role*, which acts as an interface between users and tasks saying, intuitively, "who can do what". For example, in a financial context, a *clerk* is authorized to process a loan request, but he is not authorized to sign the contract at the end of the process as only *managers* can do so. RBAC models can specify several kinds of constraints involving roles (e.g., mutual exclusivity, hierarchy, etc.), but they all fail to model constraints at user level such as, for example, the well-known *separation of duties* (*SoD*) and *binding of duties* (*BoD*) [13]. A SoD (resp., BoD) says that the users executing a set of tasks must be different (resp., equal).

Some proposals attempted at extending RBAC models to address such constraints, e.g., [12], leading to a natural question: *Does there exist an assignment of tasks to users satisfying all constraints?*, or more formally, *Is the workflow satisfiable?* If so, then it means that at least a *static plan*, precomputed before starting, to execute the workflow exists. Otherwise, either we decide not to execute the workflow or we accept that any possible execution will violate at least one constraint (and then we could look for a "least bad" plan [16] maximizing the number of satisfied constraints). Thus, the *workflow satisfiability problem* (*WSP*, [21]) is a *constraint satisfaction problem* (*CSP*) where variables model tasks and domains model authorized users. Although some techniques have been provided to solve the WSP efficiently (e.g., [17]), consistency of a CSP is NP-complete.

When an access controlled workflow does not specify any uncontrollable part workflow satisfiability is enough to synthesize a valid plan. Instead, when some part is out of control (e.g., the choice of the workflow path or the absence of users), we need, as for temporal workflows discussed above, a controllability approach to decide in real time which users to commit to which tasks. For example, consider an access controlled workflow under conditional uncertainty. In this workflow, the choices of the workflow paths to take are out of control (or by abusing language we say that these workflow paths are uncontrollable). Differently from the WSP, the assignment of a user to a task might not be precomputed before starting as the workflow may in general specify different authorization constraints for different workflow paths involving the same users for some common tasks which must be considered in any execution. In that

case, we must make this assignment while executing, typically after having full information on which workflow path we will go through (online planning).

Another controllability problem in the context of workflows with resources is the *workflow resiliency problem (WRP)*, i.e., a *dynamic* WSP coping with the absence of users. If a workflow is resilient, it is of course satisfiable, but the vice versa does not hold. A few years ago, Wang and Li defined three levels of resiliency: *static* (level 1), *decremental* (level 2) and *dynamic* (level 3) [21]. In static resiliency, up to k users might be absent before the execution starts and never become available for that execution. In decremental resiliency, up to k users might be absent before or during execution and, again, they never become available for that execution. In dynamic resiliency, up to k (possibly different) users might be absent before executing any task and they may in general turn absent and available continuously, before or during the execution. Much work has been carried out to tackle static resiliency, little for decremental and, to the best of my knowledge, nothing for dynamic.

Finally, we can of course consider variants of the previously discussed aspects. For instance, a workflow employing users and temporal constraints may specify a *temporal separation (or binding) of duties*. A temporal SoD says that a user is allowed to carry out two tasks provided a further temporal constraint is satisfied. For example, a surgeon, who has just carried out a 4-hour intervention, is allowed to do another one only after resting from 2 to 4 hours. Likewise, an aircraft pilot must rest for at least 10 hours after a transatlantic flight. These temporal constraints must be considered *in conjunction with* access control as there is a mutual influence. However, when everything is under control, these constraints boil down to normal disjunctions for which a satisfiability approach is enough. The interesting part is when some part is, again, out of control. Consider again a transatlantic flight taking off from Europe and suppose that once the aircraft lands in America, it will take off again 12 hours after the expected landing time (so potentially the same pilot is fine for the return flight). However, the exact duration of the outbound flight is uncontrollable. Suppose that it takes normally 10 hours. Once boarding is complete, the take off could be delayed for extreme weather conditions and related safety procedures such as, for example, deicing. Deicing is the process of removing snow and ice from the plane surfaces (especially wings) by "power washing" the aircraft with chemicals which also remain on the surfaces in order to prevent the reformation of the ice. If deicing process takes 3 hours (as each plane queues for its turn), the flight will land after 13 hours since boarding. As a result, the next take off will be scheduled after 9 (and no longer 12) hours, so the same pilot is not going to be fine.

To the best of my knowledge security policies involving temporal, conditional and resource uncertainty still need to be explored in depth.

When facing uncontrollable parts we can in general act in three main ways:

1. We assume that we know in advance how the uncontrollable part will behave and make sure that a (possibly different) strategy to operate on the controllable part exists.

2. We assume that we have a fixed strategy operating on the controllable part always the same way no matter how the uncontrollable one will behave.
3. We assume that we have a strategy operating (possibly differently) on the same controllable part making decisions in real time depending on how the uncontrollable part is behaving (online planning).

These are the intuitions behind the three main kinds of controllability: *weak* (for presumptuous), *strong* (for anxious) and *dynamic* (for grandmasters).

2 Contributions and Scientific Publications

Towards these unexplored directions, my contributions fall in the areas of *constraint satisfaction*, *uncertainty in AI*, *planning and scheduling*, *algorithms*, *business process management* and *security* and are the following.

1. I address temporal controllability of workflows specifying controllable and uncontrollable workflow paths and uncontrollable task durations. This part relies on temporal constraint networks. I provide *conditional simple temporal networks with uncertainty and decisions (CSTNUDs)* as a new temporal network formalism and an encoding from temporal workflows into CSTNUDs. I also address *simple temporal networks with decisions (STNDs)*, a subclass of CSTNUDs equivalent to DTNs (disjunctive temporal networks). I provide Esse (`https://github.com/matteozavatteri/esse`) and Kappa (`http://regis.di.univr.it/EE_STND2018.tar.bz2`), two tools for CSTNUDs and STNDs, respectively, with which I carry out a few experimental evaluations. The results I obtained are published in [1,5,8].
2. I address resource controllability of workflows specifying uncontrollable workflow paths. This part relies on constraint networks. I provide *constraint networks under conditional uncertainty (CNCUs)* as a new formalism of constraint networks able to model conditional uncertainty. Then, I provide an encoding from access controlled workflows into CNCUs. After that, I also address workflow resiliency via real-time controller synthesis for timed game automata. I provide Zeta (`https://github.com/matteozavatteri/zeta`) and Erre (`https://github.com/matteozavatteri/erre`), two tools for CNCUs and workflow resiliency, respectively, with which I carry out a few experimental evaluations. The results I obtained are published in [6,7,9,10,11].
3. I address temporal and resource controllability together. This part relies on further new extensions of temporal networks whose dynamic controllability is checked via controller synthesis for the corresponding timed game automata. I provide *access controlled temporal networks (ACTNs)* and *conditional simple temporal networks with uncertainty and resources (CSTNURs)* in order to model temporal security policies. I also show how the temporal constraints of a temporal role based access control model (TRBAC) can be represented as a simple temporal network to be connected to the temporal network modeling the workflow in order to understand if the access controlled workflow can be executed. The results I obtained are published in [2,3,4].

Published Papers

1. Cairo, M., Combi, C., Comin, C., Hunsberger, L., Posenato, R., Rizzi, R., Zavatteri, M.: Incorporating decision nodes into conditional simple temporal networks. In: 24th International Symposium on Temporal Representation and Reasoning (TIME 2017). vol. 90, pp. 9:1–9:17. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2017). https://doi.org/10.4230/LIPIcs.TIME.2017.9
2. Combi, C., Posenato, R., Viganò, L., Zavatteri, M.: Access controlled temporal networks. In: 9th International Conference on Agents and Artificial Intelligence (ICAART 2017). pp. 118–131. ScitePress (2017). https://doi.org/10.5220/0006185701180131
3. Combi, C., Posenato, R., Viganò, L., Zavatteri, M.: Conditional simple temporal networks with uncertainty and resources. Journal of Artificial Intelligence Research (JAIR) **64**, 931–985 (2019). https://doi.org/10.1613/jair.1.11453
4. Combi, C., Viganò, L., Zavatteri, M.: Security constraints in temporal role-based access-controlled workflows. In: 6th ACM Conference on Data and Application Security and Privacy (CODASPY 2016). pp. 207–218. ACM (2016). https://doi.org/10.1145/2857705.2857716
5. Zavatteri, M.: Conditional simple temporal networks with uncertainty and decisions. In: 24th International Symposium on Temporal Representation and Reasoning (TIME 2017). vol. 90, pp. 23:1–23:17. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2017). https://doi.org/10.4230/LIPIcs.TIME.2017.23
6. Zavatteri, M., Combi, C., Posenato, R., Viganò, L.: Weak, strong and dynamic controllability of access-controlled workflows under conditional uncertainty. In: 15th International Conference on Business Process Management (BPM 2017). pp. 235–251. Springer (2017). https://doi.org/10.1007/978-3-319-65000-5_14
7. Zavatteri, M., Combi, C., Viganò, L.: Resource controllability of workflows under conditional uncertainty. In: Business Process Management Workshops (AI4BPM 2019) (to appear). Springer International Publishing
8. Zavatteri, M., Viganò, L.: Conditional simple temporal networks with uncertainty and decisions. Theoretical Computer Science (article in press) (2018). https://doi.org/10.1016/j.tcs.2018.09.023
9. Zavatteri, M., Viganò, L.: Constraint networks under conditional uncertainty. In: 10th International Conference on Agents and Artificial Intelligence (ICAART 2018). vol. 2, pp. 41–52. SciTePress (2018). https://doi.org/10.5220/0006553400410052
10. Zavatteri, M., Viganò, L.: Conditional uncertainty in constraint networks. In: Agents and Artificial Intelligence. pp. 130–160. Springer (2019). https://doi.org/10.1007/978-3-030-05453-3_7
11. Zavatteri, M., Viganò, L.: Last man standing: Static, decremental and dynamic resiliency via controller synthesis. Journal of Computer Security **27**(3), 343–373 (2019). https://doi.org/10.3233/JCS-181244

Essential References

12. Bertino, E., Ferrari, E., Atluri, V.: The specification and enforcement of authorization constraints in workflow management systems. ACM Transactions on Information and System Security (TISSEC) **2**(1), 65–104 (1999). https://doi.org/10.1145/300830.300837

13. Clark, D.D., Wilson, D.R.: A comparison of commercial and military computer security policies. In: Security & Privacy '87. pp. 184–195. IEEE (1987). https://doi.org/10.1109/SP.1987.10001
14. Combi, C., Gambini, M., Migliorini, S., Posenato, R.: Representing business processes through a temporal data-centric workflow modeling language: An application to the management of clinical pathways. IEEE T. Systems, Man, and Cybernetics: Systems **44**(9), 1182–1203 (2014). https://doi.org/10.1109/TSMC.2014.2300055
15. Combi, C., Posenato, R.: Controllability in temporal conceptual workflow schemata. In: 7th International Conference on Business Process Management (BPM 2009). pp. 64–79. Springer (2009). https://doi.org/10.1007/978-3-642-03848-8_6
16. Crampton, J., Gutin, G., Karapetyan, D.: Valued workflow satisfiability problem. In: 20th ACM Symposium on Access Control Models and Technologies (SACMAT 2015). pp. 3–13. ACM (2015). https://doi.org/10.1145/2752952.2752961
17. Crampton, J., Gutin, G., Yeo, A.: On the parameterized complexity and kernelization of the workflow satisfiability problem. ACM Transactions on Information and System Security (TISSEC) **16**(1), 4:1–4:31 (2013). https://doi.org/10.1145/2487222.2487226
18. Posenato, R., Zerbato, F., Combi, C.: Managing decision tasks and events in time-aware business process models. In: 16th International Conference on Business Process Management (BPM 2018). pp. 102–118. Springer (2018). https://doi.org/10.1007/978-3-319-98648-7_7
19. Reijers, H.A., Mendling, J.: Modularity in process models: Review and effects. In: 6th International Conference on Business Process Management (BPM 2008). pp. 20–35 (2008). https://doi.org/10.1007/978-3-540-85758-7_5
20. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models. Computer **29**, 38–47 (1996). https://doi.org/10.1109/2.485845
21. Wang, Q., Li, N.: Satisfiability and resiliency in workflow authorization systems. ACM Transactions on Information and System Security (TISSEC) **13**(4), 40:1–40:35 (2010). https://doi.org/10.1145/1880022.1880034