Dynamic Workflow Composition with OSLO-steps: Data Re-use and Simplification of Automated Administration

Dörthe Arndt
doerthe.arndt@tu-dresden.de
TU Dresden, Computational Logic Group
Dresden, Germany

Sven Lieber
sven.lieber@kbr.be
Ghent University – imec, IDLab
Ghent, Belgium
KBR Royal Library of Belgium
Brussels, Belgium

Raf Buyle
raf.buyle@vlaanderen.be
Ghent University – imec, IDLab
Ghent, Belgium

Sander Goossens
sander.goossens@endare.com
Endare BV
Ghent, Belgium

David De Block
david.deblock@internetarchitects.be
Internet Architects
Antwerp, Belgium

Ben De Meester
Erik Mannens
ben.demeester@ugent.be
erik.mannens@ugent.be
Ghent University – imec, IDLab
Ghent, Belgium

ABSTRACT

e-Government applications have hard-coded and non-personalized user journeys with high maintenance costs to keep up with, e.g., changing legislation. Automatic administrative workflows are needed. We present the OSLO-steps vocabulary and the workflow composer: combined, they are a means to create cross-organizational interoperable user journeys, adapted to the user's needs. We identify the requirements for automating administrative workflows and present an architecture and its implemented components. By using Linked Data principles to decentrally describe independent steps using states as pre- and postconditions, and composing workflows on-the-fly whilst matching a user's state to those preconditions, we automatically generate next steps to reach the user's goal. The validated solution shows its feasibility, and the upcoming interest around interoperable personal data pods (e.g., via Solid) can further increase its potential.

CCS CONCEPTS

• **Information systems** → **Resource Description Framework (RDF)**; **Ontologies**; **Task models**.

KEYWORDS

Workflow composition; Semantics

ACM Reference Format:
Dörthe Arndt, Sven Lieber, Raf Buyle, Sander Goossens, David De Block, Ben De Meester, and Erik Mannens. 2021. Dynamic Workflow Composition with OSLO-steps: Data Re-use and Simplification of Automated Administration. In *Proceedings of the 11th Knowledge Capture Conference (K-CAP '21), December 2–3, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3460210.3493559

K-CAP '21, December 2–3, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8457-5/21/12…$15.00
https://doi.org/10.1145/3460210.3493559

1 INTRODUCTION

The EU Single Digital Gateway Regulation (SDGR) [1] encapsulates services and methodologies to guide uniform public service implementations across member states. A similar need exists on a more local scale. e-Government applications guiding users to a variety of public services ("user journeys", e.g., updating an ID card after moving across cities) are hard to implement consistently: users encounter many different online and offline (UI) tools. User journeys could be vastly simplified via automated administrative workflows. Specifically in Flanders, municipalities provide services spanning different administrative levels, increasing the implementation complexity and cost, causing them to score poorly on European benchmarks such as "government transparency" and "service delivery"[1].

The Linked Data Principles increase data interoperability [4]. Assuming that (personal) interoperable data is available for citizens, we can use this to our advantage, optimizing e-government applications to decrease the individual administrations' implementation burden.

In this paper, we describe the technical problems and present a solution for automated workflows as found within the FAST project[2]: a cooperation between Flemish academic institutes, the public sector, and industrial partners to develop automatic e-government applications optimizing user journey creation through data linking and public data, validated by a user group of different Flemish municipalities at various levels of e-government adoption.

By modeling a workflow as a series of steps, and automatically composing those steps based on the user's characteristics (the *state*), we create a system that (i) always takes the latest regulations into account, (ii) simplifies the workflow so that no duplicate data needs to be entered, and (iii) provides a basis on which multiple interactive applications can be built (HTML forms, chatbot, etc.).

After stating the requirements (section 2) we describe our architecture (section 3) and discuss the underlying data model and

[1] https://joinup.ec.europa.eu/node/702009
[2] http://project-fast.org/

workflow composition (section 4 and section 5). We validate our method in section 6, and in section 7 and section 8, we describe related work and formulate conclusions, respectively.

2 TECHNICAL REQUIREMENTS

Administrative workflows are typically hard-coded within various back-office applications, posing following limitations. First, **deprecation** by design: when regulations change, applications need to be updated, costing development time and constantly placing a burden on the technical team. During the COVID-19 crisis, almost weekly changing crisis-regulations made this limitation painfully clear. Second, focus on the **"average user"**, which does not exist: accessibility measurements and alternative interfaces (such as chatbots) are typically implemented in parallel, further adding technical burden. Third, **static** applications: the workflow is hard-coded, forcing users to re-enter the same data again and again. The Linked Data promise is automatic data reuse, but when applications follow a hard-coded flow, data reuse is limited to (semi-)automatically filling in similar data fields. Fourth, **centralized** applications, requiring centralized data: when a user journey involves multiple administrations, the user needs to access applications of those different administrations, typically needing to discover herself which is the optimal sequence.

Considering these limitations, we extract the following requirements: a workflow is (i) **automatically composed** (avoiding deprecation by design), based on (ii) **decentralized step descriptions** (uniforming interactions across multiple organizations), (iii) taking **the user's state** into account (dynamically adapting applications based on the user, i.e., the *once-only principle* [13]), and (iv) enabling multiple **alternative interfaces** (increasing accessibility).

3 ARCHITECTURE

In this section, we provide a high-level description of our workflow composition system's three components: (i) steps, (ii) the user's state and goal, and (iii) workflow composition (fig. 1). We rely on the Linked Data principles and Semantic Web technologies to ensure our descriptions are interoperable, can be published decentralized, and are unambiguous allowing for automatic processing.

Steps (fig. 1, bottom-right) are unambiguously defined descriptions of a (fine-grained) action, e.g., "provide your first name", "request a new ID card at the municipality's office", etc. They are described without stating *how* they should be executed, and independent of each other (allowing them to be published decentralized), but need some semantic connection for them to be sequenced.

The *state* (fig. 1, top) is the description of the data we could know about a specific user. A workflow is generated to change the user's original state to a specific *goal state*. The user's state is thus the semantic connection between steps and goals: by knowing a user's state, we derive which next steps we could take. For example, if a step requires to know the user's first name, we cannot take that step if the user's state does not contain its first name. We then first need to complete the step "provide your first name". By using the user's state, we can not only *skip steps that are no longer needed* (the once-only principle), we can also *personalize the workflow* providing alternatives (e.g., suggest steps tailored to people with a disability). We assume privacy and GDPR regulations are met separately.

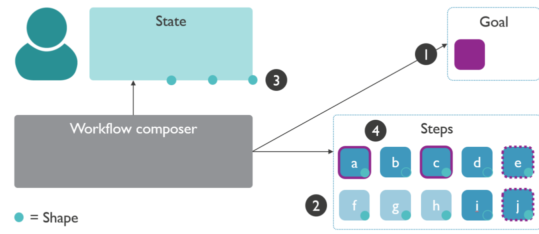

Figure 1: Architecture.
When a workflow composer (bottom-left) is triggered to reach a certain goal (top-right), it uses that goal (❶) to filter out all irrelevant steps (bottom-right, ❷, greyed out steps). Then, it matches the user state's shape (top-left) with the remaining steps' shapes (❸, circles) to return a list of possible alternative steps to take (❹, take outlined steps (a) and (c), or alternatively dashed outlined steps (e) and (j)). After a user completes the presented step(s), its state is updated and the workflow composer calculates the next steps based on the updated user state's shape (❸ and ❹) until the goal state is reached.

The *workflow composer* (fig. 1, bottom-left) composes the workflow on-the fly based on the (*latest*) available steps, the user's state, and the goal state. The result is a list of alternative next steps to take. This list of steps is described unambiguously: *alternative interfaces* can be generated on top of this description (e.g., HTML form or chatbot). When this step is completed and the user's state is updated, the workflow composer can calculate the next step to take to reach the original goal, based on the updated state.

4 DATA MODEL

Our system must be aware of the possible steps, to allow for personalized adaptive workflow composition. We developed the novel OSLO-steps vocabulary[3] to describe conditional, hierarchical steps of data-driven workflows. OSLO-steps is aligned with the P-Plan ontology[4], compatible with the Open Standards for Linked Organizations (OSLO) [5], and uses the and W3C's recommended SHapes Constraint Language (SHACL)[5]. Below, we explain and exemplify the concepts and their relation with user data.

A **Step** is an action which can be performed in a workflow, only when certain conditions are fulfilled (the *precondition*), and leading to a new situation (the *postcondition*). We describe these conditions and the resulting situation via so-called *states*. In listing 1[6], we show the `policeConfirmationAddressChange` step description (line 5), referring to a police officer visiting to confirm a citizen's address change, which is a legal requirement in Belgium if a citizen wants to register herself at a new address. This step can only be executed if the address change is declared (line 3). After the visit of the police officer, the change is confirmed, resulting in a state that can be added to the user's state (line 4).

[3] https://w3id.org/imec/ns/oslo-steps#
[4] http://purl.org/net/p-plan#
[5] https://www.w3.org/TR/shacl/
[6] For the remainder of the paper, prefixes are conform with the results of https://prefix.cc, except for
`o-persoon` for `https://data.vlaanderen.be/ns/persoon#`,
`o-steps` for `https://w3id.org/imec/ns/oslo-steps#`,
`state` for `https://example.com/states#`, and
`step` for `https://example.com/steps#`.

```
1  step:policeConfirmationAddressChange a o-steps:Step ;
2    rdfs:label "Confirmation␣of␣address␣change"@en;
3    o-steps:requiresState state:addressChangeDeclared ;
4    o-steps:producesState state:addressChangeConfirmed .
5  state:addressChangeDeclared a o-steps:State ;
6    rdfs:label "Declared␣address␣change"@en ;
7    o-steps:hasStateShape shape:addressChangeDeclaredShape .
8  shape:addressChangeDeclaredShape a o-steps:StateShape ;
9    sh:targetClass o-persoon:Inwoner ;
10   sh:property shape:addressChangeValue .
11 shape:addressChangeValue a sh:PropertyShape ;
12   sh:path ex:addressChangeDeclared ;
13   sh:hasValue true ; sh:minCount 1 .
14 :bob a o-persoon:Inwoner ;
15   foaf:givenName "Bob" ; foaf:familyName "Doe" ;
16   o-persoon:registration 123456 ;
17   schema:contactPoint [
18     schema:contactType "gsm" ; schema:telephone 8888888 ] .
```

Listing 1: Description of step "police visit to confirm a new address" with pre-/postcondition states, shapes, and user state.

A **State** represents a situation. Goals and step pre- and postconditions are described as states. Each state is resolvable via a URI, enabling us to enrich the state with additional metadata. A state has then one or more *shapes* assigned which further describes it. In listing 1 we provide a state description example. The state `addressChangeDeclared` is a state in the OSLO-steps vocabulary (line 5) and it has the shape `addressChangeDeclaredShape` associated to it as a shape (lines 7–10).

A **Shape** specifies how the data of a state looks like. This is needed to align the user's state to potential steps: "can we match the shape of the user's state with the shape of the preconditions of the available steps?". Since we align with OSLO and the Linked Data principles, we need a shape description for data described using the Resource Description Framework (RDF). Hence, we use SHACL to indicate which data needs to be present in a state description. Using SHACL allows the workflow composer to determine the required data structure, and the expected kinds of values (e.g., strings following a certain pattern, integers being within a range). An example shape is displayed in listing 1: in order to obtain the state the shape is describing, there needs to be a triple with the predicate `ex:addressChangeDeclared` (lines 11–12) and the object `true` (line 13) for each instance of the class `o-persoon:Inwoner`.

The step, state, and shape descriptions can be used by our workflow composer to create generic workflows. By relying on the **user state**, we can compose a personalized workflow, i.e., optimal steps and step sequences per user. We assume that this data resides in one or more trusted locations, accessible to the workflow composer. Indeed, by relying on the Linked Data principles, we can make use of a decentralized network of personal data which can be federated to compose an optimized workflow. The user state determines which information is already present and which requirements are already fulfilled, and hence which steps are unnecessary to execute. Listing 1 shows information of a user (lines 14–18). Following the Linked Data Best Practices, we maximally rely on established vocabularies such as `foaf` or `schema.org`[7]. In the example, we already have the name, the national number, and the cell phone number of the user and can therefore omit the steps which are in place to acquire that information. Considering the example of listing 1, the step `step:policeConfirmAddressChange` is not necessary if the profile of the user already contains the information that the address change has been confirmed by the police.

[7] http://xmlns.com/foaf/spec/ and https://schema.org/, respectively.

5 WORKFLOW COMPOSITION

We need dynamic workflow composition to adhere to the aforementioned requirements. For this, we relied on a semantic workflow composer which was originally developed for a healthcare set-up [10]. The original composer uses weighted state transition logic, a logic to model state changes based on actions planned in clinical pathways. As such, the composed workflows to reach user-defined goals are adapted to the available set of actions, the current state of a patient, and optimized by applying weights on the actions. We extended this composer for e-government workflows.

The composer is implemented using Notation3 logic, in the N3 language [3], using the gps-schema ontology[8]. Each step is expressed as a rule and consists of a *from-part* containing the data which becomes false by the execution of the step, a *where-part* comprising the requirements which need to be met for the step execution, a *to-part* expressing the expected result of the step execution, and some *weights* to express quantitative parameters for each step. The direct implementation in Notation3 logic makes it furthermore possible to add extra reasoning rules which can be executed during the workflow composition to, e.g., do small calculations or express equivalencies between different vocabularies present in the user state. The composition of the different possible workflows is done with the EYE reasoner which was chosen because of its expressive power, its support of built-in functions and its strong performance [12].

To align with vocabularies established in e-government in Flanders such as the OSLO standards and SHACL – essential to allow for adoption – we chose to translate the OSLO-steps descriptions to the internal data model used by the workflow composer instead of using it directly. For easy integration with other systems, the result was wrapped in a RESTful web API demonstrator[9]. This API allows to upload new step, state, and shape descriptions, optimize the composer for a given goal, and compose workflows adapted to the user state given as input.

6 VALIDATION

The OSLO-steps descriptions and workflow composer were validated by the FAST project consortium using the onboarding life event: a citizen moving from one (Flemish) city to another, requesting, e.g., an address change on its ID card, the right types of municipality trash cans, and a police visit to confirm the move. This use case was chosen as it was no trivial use case, requiring dependent and independent steps with overlapping data requirements, and a mix of online and offline actions that need to be taken to complete the workflow (e.g., changing the ID card address can be done online, the police confirmation requires an offline visit). The entire validation was demonstrated live and its recording is available online[10].

[8] http://josd.github.io/eye/reasoning/gps/gps-schema
[9] https://w3id.org/imec/oslo-steps/api/docs
[10] http://project-fast.org/#closingevent, "FAST demonstrator and valorisation"

Applications were built by different partners to validate its use, using the step, state, and shape descriptions, and using the output of the workflow composer. For example, when a new user starts an application, the workflow composer is triggered, and its result contains, e.g., "provide first name" and "provide last name", with identifying links to the respective step descriptions. Using the step and linked state and shape descriptions, a user interface was automatically generated, e.g., a form with two text fields. However, the same information allowed to generate alternative user interfaces, such as a chat bot. As the shape descriptions also describe the expected kinds of values, the user interfaces could automatically include validation mechanisms, e.g., validating your phone number is entered in the right format, your birth date is after a certain date, etc.

Additionally, a visual editing environment was created to adjust the different step, state, and shape descriptions. This improvement in user experience could increase uptake among government workers, and was used to showcase that updating the different descriptions had an immediate effect on the composed workflows and on the consequently automatically generated user interfaces. E.g., when removing the state "provide first name" as a possible precondition, the automatically generated user interface no longer includes a text field for entering the first name.

Endare used the findings of FAST, and further applied them to a non-government context. The concept of generating personalized forms using the workflow composer was applied to the AquaFlanders and TMCS projects[11]. Both applications use these results to generate complex inspection forms, personalized based on the user and the inspection at hand.

7 RELATED WORK

We focus in our related work section on data-driven workflows as a possible solution to our requirements, given its more natural alignment with the graph-based data structures of Linked Data. This excludes control-driven approaches, e.g., using Business Process Model and Notation (BPMN)[12]. Although the base concept of a step containing pre- and postconditions is prevalent in literature across different use cases [6, 7], workflow composition systems that rely on them are typically not adaptive, i.e., they do not start from the current context – in our case, the user state – and rely on the input of a concrete starting point instead [8, 9]; or are applied to web service composition, i.e., they cannot take the combination of online and offline tasks into account [2, 11]. Generic vocabularies to describe pre- and postconditions exist, e.g., the recently published CCCEV 2.0 vocabulary[13], to which OSLO-steps could align and extend to include workflow composition.

8 CONCLUSION AND FUTURE WORK

We described the requirements needed for an adaptive workflow composition system, and introduced how OSLO-steps and the workflow composer fulfil these requirements to create multiple alternative optimized workflows taking the user state into account. Our decentralized approach is specifically useful in Flanders where multiple governmental agencies need to cooperate on different organizational levels, but could further be beneficial in many use cases. We plan to include business rules to automatically derive data when composing workflows (e.g., based on the user's birth date we can derive whether maturity is reached or not) and further expand the possibilities of OSLO-steps and the workflow composer.

The importance of personal user data control increases, evidenced by the uptake of Solid within the Flemish government[14], and we can envision our solution to be more easily integrated in existing systems, as our assumption of having interoperable, GDPR-compliant personal data pods becomes closer to reality every day.

ACKNOWLEDGMENTS

The described research activities were funded by Ghent University, imec, and Flanders Innovation & Entrepreneurship (VLAIO). We especially thank our FAST partners: CrossLang, Endare, Ideabox, Digitaal Vlaanderen, Internet Architects, and Vlerick.

REFERENCES

[1] 2018. Regulation (EU) 2018/1724 of the European Parliament and of the Council of 2 October 2018 establishing a single digital gateway to provide access to information, to procedures and to assistance and problem-solving services and amending Regulation (EU) No 1024/2012 (Text with EEA relevance.). *Official Journal of the European Union* (2018).

[2] Ali A. Alwasouf and Deepak Kumar. 2018. Research Challenges of Web Service Composition. In *Advances in Intelligent Systems and Computing*. 681–689. https://doi.org/10.1007/978-981-10-8848-3_66

[3] Tim Berners-Lee, Dan Connolly, Lalana Kagal, Yosi Scharf, and Jim Hendler. 2008. N3Logic: A logical framework for the World Wide Web. *Theory and Practice of Logic Programming* 3 (2008), 249–269. https://doi.org/10.1017/S1471068407003213

[4] Raf Buyle, Mathias Van Compernolle, Dieter De Paepe, Jens Scheerlinck, Peter Mechant, Erik Mannens, and Ziggy Vanlishout. 2018. Semantics in the wild: a digital assistant for Flemish citizens. In *Proc. of the 11th Intl. Conf. on Theory and Practice of Electronic Governance*. https://doi.org/10.1145/3209415.3209421

[5] Raf Buyle, Laurens De Vocht, Mathias Van Compernolle, Dieter De Paepe, Ruben Verborgh, Ziggy Vanlishout, Bjórn De Vidts, Peter Mechant, and Erik Mannens. 2016. OSLO: Open Standards for Linked Organizations. In *Proceedings of the International Conference on Electronic Governance and Open Society: Challenges in Eurasia*. 126–134. https://doi.org/10.1145/3014087.3014096

[6] Ben De Meester, Tom Seymoens, Anastasia Dimou, and Ruben Verborgh. 2020. Implementation-independent Function Reuse. *Future Generation Computer Systems* (2020), 946–959. https://doi.org/10.1016/j.future.2019.10.006

[7] J. Ison, M. Kalas, I. Jonassen, D. Bolser, M. Uludag, H. McWilliam, J. Malone, R. Lopez, S. Pettifer, and P. Rice. 2013. EDAM: an ontology of bioinformatics operations, types of data and identifiers, topics and formats. *Bioinformatics* 10 (2013), 1325–1332. https://doi.org/10.1093/bioinformatics/btt113

[8] Vedran Kasalica and Anna-Lena Lamprecht. 2020. APE: A Command-Line Tool and API for Automated Workflow Composition. In *Proceedings of Computational Science – ICCS 2020*. 464–476. https://doi.org/10.1007/978-3-030-50436-6_34

[9] Magnus Palmblad, Anna-Lena Lamprecht, Jon Ison, and Veit Schwämmle. 2018. Automated workflow composition in mass spectrometry-based proteomics. *Bioinformatics* 4 (2018), 656–664. https://doi.org/10.1093/bioinformatics/bty646

[10] Hong Sun, Dörthe Arndt, Jos De Roo, and Erik Mannens. 2021. Predicting future state for adaptive clinical pathway management. *J. Biomed. Informatics* (2021), 103750. https://doi.org/10.1016/j.jbi.2021.103750

[11] Ruben Verborgh, Dörthe Arndt, Sofie Van Hoecke, Jos De Roo, Giovanni Mels, Thomas Steiner, and Joaquim Gabarró. 2017. The Pragmatic Proof: Hypermedia API Composition and Execution. *Theory and Practice of Logic Programming* 1 (2017), 1–48. https://doi.org/10.1017/S1471068416000016

[12] Ruben Verborgh and Jos De Roo. 2015. Drawing Conclusions from Linked Data on the Web: The EYE Reasoner. *IEEE Software* 5 (2015), 23–27. https://doi.org/10.1109/MS.2015.63

[13] Maria A. Wimmer. 2021. Once-Only Principle Good Practices in Europe. In *The Once-Only Principle*. Springer International Publishing, 61–82. https://doi.org/10.1007/978-3-030-79851-2_4

[11] https://www.endare.com/project/fast/

[12] https://www.bpmn.org/

[13] https://semiceu.github.io/CCCEV/releases/2.00/

[14] https://overheid.vlaanderen.be/informatie-vlaanderen/nieuws-en-agenda/solid-ecosysteem