

A semantic-based approach for automating compliance by the design of digital services - a case study in the academic sector

*Giovanna Di Marzo Serugendo*¹, *Ashley Caselli*^{1,*}, *Maria Assunta Cappelli*¹, *Lamia Friha*², *Alain Hugentobler*², *Kerfalla Cissé*¹, *Paul Mulard*¹, *Nikita Missiri*¹, *Audrey Martelli*¹, *Bernard Huyhn*¹, *Ismet Cem Turhan*¹, *Obaydollah Sadeghi*¹, *Saleh Bauch*¹, and *Ahmed Buzgheiba*¹

¹Centre Universitaire d'Informatique, University of Geneva, Route de Drize 7, CH-1227 Carouge, Switzerland

²DISTIC, University of Geneva, 24, rue Général-Dufour, CH-1204 Geneva 11, Switzerland

Abstract. Monitoring University students' progress, in an interactive, synchronised, and coordinated way across a given study program is a challenge. Indeed, for a given student journey, University actors need a service offering personalised views and actions depending on their role (students, study advisors, scientific committee, faculty members, admission office), all linked to the same underlying information system, and abiding study regulations. The challenge is even greater when it comes to scaling up and providing such service for the whole University community (approx. 23K members in Geneva). Following a service science approach, we derived a method for automatically generating a service, compliant by design to study regulations, and offering personalised views to the various actors. The automatic implementation of the study regulations is based on the definition of generic rules able to define the different elements of the regulations that need to be respected across all curricula and programs. We provide a case study based on the PhD students regulations. We describe a proof-of-concept instantiating the service from the rules, role-centric views of the interface, as well as the underlying architecture relying on a semantic reasoning engine. The work presented in this paper has the potential to alleviate and improve the tasks of all the various actors involved in students monitoring, going beyond PhD students. This approach automatise the implementations of any study program and can be applied to any University.

1 Introduction

During the last five years, the University of Geneva (Unige) set up an innovation process aiming at developing innovative digital services for the University. In a previous paper [1] we reported on this unique process, that follows a service science approach, and where researchers, students, information systems experts and stakeholders team up to ideate services, develop ideas into functional prototypes, and industrialise and deploy them for the whole University community. This innovation process is a funnel, where some of the produced

*e-mail: Ashley.Caselli@unige.ch

ideas are transformed into prototypes and the most promising prototypes are turned into an industrial service.

As part of this innovation process, and during a first innovation session, we developed a full roadmap for automating the instantiation of study regulations and study plans [2]. Indeed, with more than 300 programs and 19'000 students, there is a strong interest, among the various stakeholders, in a *digital service* being able: (1) *to provide multiple views of the students' progress in a program* (view from the student, the study advisor, the program director, the Faculty, etc.); (2) *to automatically instantiate the corresponding study plan*; and (3) *to automatically ensure compliance by design with the corresponding study regulations*.

Programs are defined by their study regulations and study plan (courses), and both vary in their structure and success conditions. Additionally, specific conditions apply to individual students (personalisation), and yearly changes to a given study regulation and study plan lead to cohorts with different programs. The challenge, and primary research question, is *how to provide a generic process that applies to all programs, leading to a digital service compliant by design with regulations, accommodating the variety of cases, study plans and conditions, including personalisation and study regulations changes*.

The main idea at the basis of our proposal lies in: (1) the definition of a *meta-model for expressing generic rules* in which we can frame all the elements of a study regulation; (2) a *semantic reasoning engine* capturing the rules of a specific program; (3) a *digital service* providing coordinated role-based actions of the various actors, relying on the semantic reasoning engine to ensure compliance with the regulations. We thus aim to set up a generic method to automate the implementation of any study regulations and study plans and to provide a service, for monitoring students progress, that is compliant to those regulations. We report on our latest advances on this topic, resulting from a second innovation session on November 4th and 5th 2022, and from a student group coursework taking place in Autumn 2022.

Section 2 discusses related works. Section 3 highlights our work under the service science perspective. Section 4 describes our case study and the expected business process we intend to set up. Section 5 describes the generic rules enabling the instantiation of the study regulations. Section 6 explains their instantiations, while Section 7 describe their semantic-based implementation. Section 8 discusses role-centric views of the service. Finally, Section 9 concludes the paper.

2 Related works

Business Process Automation. Business processes in the education sector are designed to manage the processes such as the education offer, curricula, recruitment, enrolment for students and staff registration, etc. Lautenbacher et al. [3] show the benefits of an ontological grounding for the business process model (BPM). They explain the semantic annotations of BPM and the use of ontologies makes BPM become not only machine-readable but also "machine-understandable". Fonou Dombeu et al. [4] propose an infrastructure that integrates Business Process Modelling (the discipline that designs, models, and optimises government processes) and Semantic Annotation (that adds semantic descriptions to process models with the ontology) into existing Semantic Web Services (SWS) solutions as tools to model and engineer SWS for non-automated government operations and processes. Born et al. [5] propose a semantic business process management approach. Their proposal consists of allowing the annotation semantics in a user-friendly way. They expose ontological knowledge to the business user in appropriate forms and by employing matchmaking and altering techniques to display options with high relevance only. Bala et al. [6] develop SHAPEworks in the context of a BPMS scenario. SHAPEworks is a prototype that demonstrates the benefits of an integrated solution of different approaches implemented on top of a real Business Process

Management System (BPMS), such as automatic reasoning, ontology and process mining. The reasoner module is also in charge of validating SHACL constraints for potential violations of domain constraints.

Compliance checking. Besides automating processes, it is also important in some cases to provide a service that is compliant to internal policies, regulations or legal rules. Compliance checking refers to the process of verifying that a digital service adheres to a set of normative requirements [7]. Compliance checking can be done manually or in an automated manner at run-time. Among run-time compliance checking, some proposals involve checking the service against formal specifications of the regulations, while other proposals involve ontology and semantic aspects. El Kharbili et al. [8] work on semantic compliance management. Their approach includes modelling compliance measures based on policies and providing a framework to manage and enforce compliance policies on enterprise models and BPM. Sapkota et al. [9] propose a service, exploiting semantic Web technologies that support the management of a compliance system. This service automates the extraction of regulatory information from regulatory texts and maps regulations to organisational processes. It does not provide automated reasoning exploiting those rules for compliance checking. Beach et al. [10] provide a generic rule-based semantic regulatory compliance checking methodology, applied to the construction domain. Compliance checking consists in processing SWRL rules, expressing the regulations, on a building data, de facto checking the compliance of the building to the regulations.

Compliance by design. Compliance by design refers to digital services where normative requirements are taken into account during the modelling, or during the design of the service [11]. Lohmann [11] provides an approach, where Petri nets specify artifact-centric business processes, including the specification of the rules the business process must comply with. As a consequence, implementing the software in accordance to the specification leads to a software that is compliant by design.

There are still no well-defined BPA solutions that combine automatic reasoning methods in the education domain and leverage the advantages of semantic technologies to check institutional regulatory compliance. Proposals involving semantic Web technologies, or rule-based semantic compliance, sharing similarities with our approach, focus on verification of compliance (compliance checking or compliance management), while we aim at providing in an automated manner a digital service that is by design compliant with study regulations.

3 Service science approach

The work described in this paper follows a service science approach, as it displays the following characteristics identified in service science activities [12].

- *Services: co-creation of value and innovation.* Stakeholders are fully involved in the co-creation of the service we developed. We first started the work on the study regulations during a hackathon in 2021 [2] and continued the work in 2022. We involved study advisors, program directors, students and information systems experts in the implementation of study regulation. They provided knowledge related to the regulations and to the role-centric views and functionalities required to monitor students progression.
- *Service system.* We are clearly at the center of a system made of *people* (academic community - students, administrative staff, teachers), *businesses* (students' offices, study programs), and *technology* (information system, e-learning services, etc.).
- *Interdisciplinarity.* Key to our approach, interdisciplinarity is at the heart of the innovation process we set up at Unige [1]. Indeed, this process brings together researchers and IT professionals, includes various experts of specific domains brought together in the co-creation

of the respective services (e.g. experts in Unige teaching programs regulations as discussed in this paper, or experts in e-learning platforms, experts in delivering sport and culture to the University community, health experts for health and well-being services).

4 Case study: PhD students

4.1 Case study

There is a high interest from the various stakeholders in monitoring students' progress across a given study program. There is a pressing need to start with PhD student programs as so far there is no digital service providing any monitoring for those students at Unige.

The *AS-IS process* is manual and essentially uses spreadsheets to track PhD students progress. Study advisors' maintain those files for the various PhD programs and update them manually. Similarly, for PhD students, it is impossible to verify the status of their application (during the admission process), whether their thesis subject has been accepted, or whether their yearly progress report is approved. As a consequence, PhD students have no clear view of their own progression and of exact deadlines. They need to refer to the admission letter and to the study regulations to perform themselves all deadlines. Finally, for PhD thesis supervisors or PhD program directors, there is no possibility to check the status of their PhD students either during the admission progress or while they are enrolled in the program.

The *TO-BE process*, co-created and defined with the stakeholders, in particular the study advisors, PhD program directors and the PhD students, needs: (1) to be *automated* with clear identification and visualisation of students' progression. A study advisor is interested in being able at a glance to identify all PhD students being delayed in a given program (e.g. not enough credits acquired at a certain semester, missing yearly progress reports, thesis subject not submitted within the first two semesters of study, etc.). PhD students are interested in identifying incoming tasks and deadlines. PhD thesis supervisors or PhD program directors want to monitor their own students or those enrolled in the program they are responsible for; (2) to *integrate study regulations* and corresponding tasks and deadlines for each student, which depend on the study regulations and the date the PhD student enrolled; (3) to provide a service with *role-based actions* through which multiple actors can interact. For instance, PhD students submit their applications through the service, which then informs the study advisor about a new application, that she can transfer through the service to the PhD program director, who accepts or rejects the application.

4.2 Approach

A first functional prototype attracted interest from the stakeholders and comforted us to go further and provide a complete service. We started from the roadmap we identified previously [2] expressing the need to: (i) model study regulations and study plans as a set of basic building blocks; (ii) provide an ontology and reasoning rules to enforce the constraints of the study regulations; (iii) provide interactive visual tools for defining study regulations, study plans and students personal programs; (iv) develop a role-centric service with different views for monitoring students.

As the result of a first ideation process, we provided an ontology and a preliminary set of reasoning rules [2] expressed in SHACL (point (ii) above). During a second ideation process, we focused on points (i) and (iv) above and describe them in this paper. We specifically worked on the following points:

1. Modelling *generic rules* that correspond to building blocks discussed in our roadmap [2]. These rules are generic and are able to accommodate all programs.

2. *Identifying rules*, to implement, specific to the PhD study regulation and study plan.
3. *Instantiation of the generic rules* to describe the PhD study program regulation and study plan, and express them in a specific format (e.g. JSON, XML or RDF).
4. *Implementation of the rules* providing de facto an automatic instantiation of the study program (e.g. with SHACL¹).
5. *Development of a digital service* for managing students *automatically compliant* with regulations derived from the instantiation and implementation of the generic rules.
6. Providing *personalised views* (for students, study advisor, program director).
7. Allowing *coordinated role-based actions* of the actors through the same service.

The advantage of such as service built around multiple actors (student, thesis director, study advisors, program director), interacting jointly on the same service are multiple:

- Everyone can see the progress of individuals or groups of students at any time;
- Information is centralised and available at the same place;
- The approach provides a role-centric view, where all stakeholders can see the same process and can act on it according to their role;
- Everyone can take the actions corresponding to their role and they are immediately visible to the people concerned;
- Optimisation of the management of individual students or a whole cohort;
- Facilitates individual student path and provides them with an optimal UX.

We leveraged a group of students from the BSc in Information systems and service science to provide a functional prototype. They used an agile methodology with regular sprints to progress with their assignments. As part of their assignment, they needed to gather the needs of the various stakeholders, provide mockups of the interface, and implement the database as well as the back- and front-end. To further accelerate the work, we participated together with the students in a two-days hackathon on November 4th and 5th, 2022.

4.3 Process and regulations

We describe here the TO-BE process as we envisage it. A PhD prospective candidate submits her application file. The Unige admission process applies a screening process that either leads to the application being rejected (e.g. the candidate has no MSc degree) and informs the candidate accordingly, or the application is validated and passed on for further screening by the Faculty in charge of the PhD program requested by the candidate. In this case, the Faculty admission (e.g. the study advisor) sends a preliminary decision to the Scientific Committee of the program which will further formally accept or reject the application. In both cases, the decision is transmitted to the Faculty admission, which informs the candidate of the success or not of her application. In case the application is accepted, the candidate proceeds to admission and enrolment. A Unige student database records all the various actions and decisions and stores the corresponding files (e.g. application and copies of the diploma).

Once PhD students are enrolled, they have a maximum of two semesters for: (i) taking courses they are asked to attend; (ii) submitting the thesis subject (a 4/5 pages document describing the PhD topic, research questions and method). In case they fail the courses, they are eliminated from the program. The student supervisor and the PhD program director assess the thesis subject and send it to the Faculty, who will further accept or reject the thesis subject. In case of rejection, the student provides a new version. Each year, the student must provide

¹<https://www.w3.org/TR/shacl/>

a progress report that is evaluated by the scientific committee and sent to the Faculty, who informs the student about the acceptance or not of the progress report. A rejected progress report may lead to the elimination of the student from the program. Towards the end of the process, the PhD student submits the manuscript and goes to a PhD thesis defence, where she may fail or succeed. All actions, decisions and files are stored in the Unige student DB.

5 Generic rules to model regulations

This section discusses the generic rule meta-model that we designed to represent the study programme regulations of the University of Geneva. Regulations are composed of several articles. Each article defines a *rule* that some *actors* must comply with. To comply with a rule an actor must carry out some task or take an *action*, often within a certain *constrained* period of time. Thus, the core of the model is built on the following concepts: Rule, Actor, Action, and Constraint.

We define a Rule as an entity that specifies an action that an actor must fulfil within the defined constraint.

We define an Actor as an entity that, according to their role, is asked to perform one, or more, actions. We then define an Action as an action done by a subject, namely the actor. The Action is in turn a composition of: (i) a *type* that describes the type of the action (i.e. attend, submit); (ii) an *object* that describes the object of the action (e.g. event, document); (iii) the *expected resource*, if applicable, that provides information about the expected resource (e.g. when submitting a report we might expect a PDF resource); and (iv) the *duration*, if applicable, that specifies how long the action must last.

In addition, we define a Constraint entity by its type (e.g. temporal) and some properties that are dependent on the constraint type. In the case of a temporal constraint, for instance, it is represented as an interval with a beginning (*start*) and an end property. Both start and end properties are in turn represented using a temporal *indicator* (i.e. every, until, before) and a *when*, that specify the date from or until, the constraint is valid.

Listing 1 shows the EBNF grammar that formally defines the rule meta-model. For the sake of simplicity, we only show the grammar syntax that can be used to represent the regulations related to a PhD Student.

In order to ease the readability, we here provide an example using a visual notation of the above-presented grammar. The example presents some articles extracted from the regulations of the PhD programme in Information Systems of the University of Geneva. Together with the articles we also provide their representation using the rule model described above. The articles we selected are the followings:

- **Core courses:** during the first year, PhD Students need to take a core course on research methods, scientific writing, and presentation skills (Design Science Research course or DSR course) amounting to 2 credits. During the first three years, PhD students need to complete: a written literature review and orally present three scientific articles (State of the art course); a written scientific article on their research (Writing Scientific Paper course).
- **Advanced scientific seminars:** Within the five years of their PhD, but before discussing the thesis, PhD Students must attend at least 5 days of advanced scientific seminars, academic summer schools, workshops, or conferences on a relevant information systems topic.
- **Annual progress evaluation:** Each year, PhD students provide a progress report and orally present their work to the PhD scientific committee, who will validate the progress and the continuation of the PhD.

Figure 1 displays the first part of the articles named “Core courses” represented using the generic rule model we designed. Doing that we assume that the knowledge about what

```

Rule      ::= '{ "actor" : Actor, "action" : Action, "constraint" : Constraint }';
Actor     ::= "PhDStudent" | "Study advisor" | "Supervisor" |
             "Program director" | ... ;
Action    ::= Attend | Submit | Validate | ... ;
Constraint ::= Temporal_constraint | ... ;
Attend    ::= '{ "type" : "attend", "object" : Event,
               [ "duration" : Time_duration ] }';
Submit    ::= '{ "type" : "submit", "object" : Document,
               [ "resource" : Expected_resource ] }';
Validate  ::= '{ "type" : "validate" }';
Event     ::= "conference" | "course" | "seminar" | ... ;
Time_duration ::= [0-9]* ("days" | "weeks" | "months" | "years" | "semesters" | ... );
Document  ::= "report" | "thesis subject" | ... ;
Expected_resource ::= ".pdf" | ".tex" | ".zip" | ... ;
Temporal_constraint ::= '{ "type" : "temporal",
                          '{ ["start" : Time_constraint], "end" : Time_constraint }';
Time_constraint ::= '{ "indicator" : Indicator, "when" : When }';
Indicator  ::= "from" | "until" | "every" | "within" | "after" | "before" | ... ;
When       ::= Step | Date | Time_duration ;
Step       ::= "enrolment" | "defence" | ... ;
Date       ::= DD/MM | DD/MM/YYYY ;

```

Listing 1: EBNF grammar that defines the rule model syntax

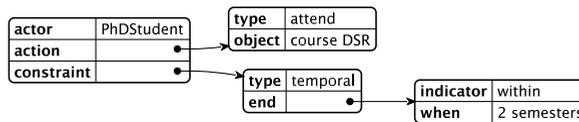


Figure 1: “Core courses” regulation represented through the generic rule model

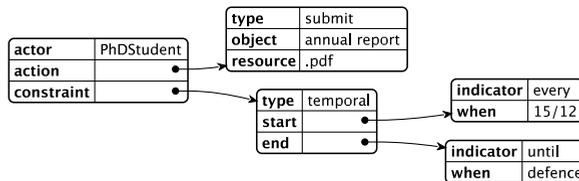


Figure 2: “Annual progress evaluation” represented through the generic rule model

“course DSR” is already known in the system, and also that the starting date is implicitly known (i.e. PhD enrolment date). Figure 2 provides a representation of the rule referred to as “Annual progress evaluation”. Finally, Figure 3 shows the “Advanced scientific seminar” article represented through the model. In this case, it uses the duration field of the action in order to represent how long the action must be (5 days).

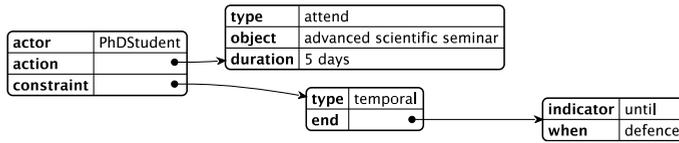


Figure 3: “Advanced scientific seminars” regulation represented through the generic rule model

6 Instantiation of the rules

Every time a student enrolls on a study programme there are regulations that they must comply with. Different entities may issue such regulations at different levels: university level, faculty level, department level, study programme level, etc. In addition to the different regulations to comply with, in some study programmes, there might be no strict admission deadline, thus admission requests are accepted throughout the year. This loosened admission process entails more freedom and flexibility, resulting in a more challenging regulations checking process. Indeed, every student will result in having to perform the same steps at the same time, but with different deadlines. This means that every time a student completes the enrolment and admission process, customised study regulations must be applied with the deadlines customised for the specific case. It is the case of the PhD students at the University of Geneva, that once enrolled must have a customised instance of the regulations since they share only a few deadlines. In the following section, we provide an example of the regulations.

6.1 Student enrolled on the PhD in Information Systems programme

Starting with the articles extracted from the University of Geneva regulations mentioned above, we here provide an example of the customs regulations that are instantiated for a particular PhD student. From here on we make the assumption that there exists a new PhD student (with student id = 209384) in the system that has just completed the enrolment process. Also, to ease the example we assume that the enrolment process was completed on the first day of the winter semester of the academic year 2022/2023, namely on the 19th of September 2022. The customised regulations the student will be subject to are the followings:

- **Core courses:** the student has to take the mandatory classes by the 18th of September 2023 (during the first year or within 2 semesters starting from the enrolment date).
- **Advanced scientific seminar:** the student has to attend at least 5 days of advanced scientific seminars, academic summer schools, workshops, or conferences on a relevant information systems topic between the enrolment date (September 19th, 2022) and before the end of their PhD.
- **Annual progress evaluation:** every year the student has to provide a PhD progress report by the 15th of December (the date is the same for all students, required by the Faculty).

Such regulations are then instantiated and represented using a model that resembles the meta-model presented above. Indeed, the model is taken as a sort of template and the information is replaced with the actual data. However, there might be cases in which the instantiation is only partially done. This is because there might be some information that is not yet available at the instantiation time, thus it is needed to keep the meta-information together with the instantiated data. Once the information becomes available, the meta-information will be replaced with the actual data (i.e. normally the thesis defence date is not available before the last 12 months

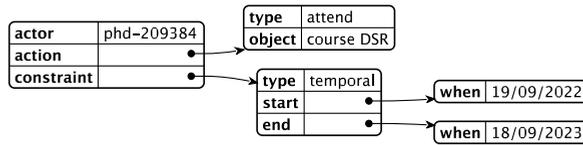


Figure 4: Instantiation of the rule called “Core courses” with the actual student id, and start and end date

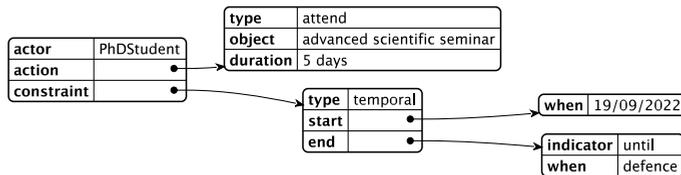


Figure 5: Partial instantiation of the rule called “advanced scientific seminar” with the actual start date and keeping the meta-information about the end field until it is not available

of the PhD). Figure 4 shows the instantiation of the rule called “Core courses” (shown in Figure 1). During the instantiation of the rule, the regulation represented through the meta-model will be read and replaced with the actual data. In this case, as Figure 4 shows, the actor was replaced with the specific student (id 209384), and the start and end dates were replaced with the actual dates. Regarding the action object “course DSR”, in this case, we assume that somewhere in the system there exists such knowledge and that it is uniquely identifiable. Another example of instantiation is provided by Figure 5 which shows the instantiation of the “Advanced scientific seminar” rule. In this case, the instantiation adds a start date which corresponds to the student enrolment date but keeps the end field with the same information. This is indeed the case where the information is not yet available, and most likely it will not be available until before the last 2 semesters of the PhD journey. Once the defence date is set, it will be then possible to replace, namely instantiate, the end field with the actual information.

6.2 Compliant PhD monitoring service

Figure 6 shows a full overview of the PhD monitoring service and the different elements that render it automatically compliant with regulations and study plans.

The generic rules meta-model describes the concepts of Rule, Actor, Action and Constraint defined in Section 5. These rules are defined for the specific case of the PhD study regulation program and study plan as shown in Figures 1 to 3. An underlying ontology for study regulations and study plans supports a reasoning tool, and a triple store (all discussed in [2]). The triple store contains both instantiated information regarding the students (coming from the PhD students DB), as well as instantiations of the rules as shown in Figures 4 and 5. The reasoning engine contains the actual implementation of the rules (Section 7). By applying the rules on the students information present in the triple store, the reasoning engine provides the status of each student to the various role-centric views (e.g. on track, delayed, etc.). The remainder of the service consists of the various views that: draw information from the triple store to display the up-to-date view; allow coordinated role-based actions; and modify the triple store accordingly once an action has happened (Section 8).

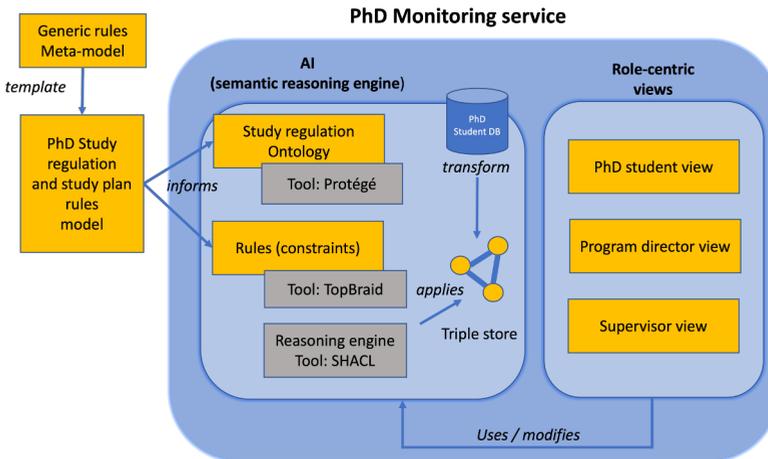


Figure 6: General overview of the PhD monitoring service - including compliance to regulations and study plan

Personalisation and yearly changes. PhD students are subject to different regimes, they may be asked to attend classes in line with their own PhD subject. Following on the recent pandemic, the University agreed to grant an additional semester (going beyond the regulation) for those students who had been slowed down in their research because of the pandemic. There is a clear need for personalisation of the program, as it emerged from the requirements discussions with the stakeholders. Through her role-centric view, the study advisor defines personalised aspects for each student, when necessary. These elements, if linked to the regulations, will supersede the original rule that served to create service. The triple store data for that student is updated accordingly.

Besides, study plans and regulations change frequently, practically on a yearly basis and there is a need to facilitate regulations change that must apply to entire cohorts. Different cohorts with different regulations for a given study program will have their own specific rules expressing their study regulations (as if it was another program). If a program change affects students previously enrolled, this will be treated as a personalisation.

7 SHACL-based Rules

We convert two of the above generic rules through SHACL. We create some classes into TopBraid², and then some SHACL nodes are coded for each interested class. Each node shape is connected to the corresponding class through SHACL target class. A target class allows that the definition of a node shape applies to all instances of a class. Finally, we add the relevant features as part of a node shape by pointing out the relevant features as property shapes using the `sh:property`.

Rule Core Courses: A PhD student attends the DSR course within 2 semesters

Listing 2 declares a `unige:PhDStudentShape` with two properties. The first property is that a PhD student `unige:attends unige:CourseShape`. The second property enunciates the value that `unige:TimeToAttend` must have. This constraint checks that the property has the value `unige:within2semesters`.

²TopQuadrant, Inc. developed TopBraid. We use TopBraid Composer Maestro Edition version 7.1.1.

We set the instance `unige:Roxane`, who is a PhD student. Figure 7 shows the following outcomes: `unige:Roxane` attends the Design Science Research (`unige:DSR`) within 2 semesters. The rule passes the validation.

```
unige:PhDStudentShape
  rdf:type sh:NodeShape ;
  sh:property [
    sh:path unige:attends ;
    sh:node unige:CourseShape ;
  ] ;
  sh:property [
    sh:path unige:timeToAttend ;
    sh:hasValue unige:within2semesters ;
  ] ;
  sh:targetClass unige:PhDStudent ;
.
```

Listing 2: SHACL-Based Rule - Core courses

Resource Form

Name: unige:Roxane

Annotations

Other Properties

- unige:attends
- unige:DSR
- unige:depositStartDate
- unige:endData
- unige:submits
- unige:timeToAttend
- unige:within2semesters

rdf:type

- unige:PhDStudent

Incoming References

Form | Source Code

Imports | Instances | Domain | Change History | Error Log | S

[Resource]	rdf:type
unige:Roxane	unige:PhDStudent

Figure 7: The instantiation of SHACL-based Rule Core Course (pass validation)

Rule Advanced scientific seminars: PhD student attends advanced scientific seminars for 5 days before the date of the thesis defence

Listing 3 defines a `unige:PhdStudentShape` with three properties. The first property lays down that a PhD student `unige:attends` `unige:AdvancedScientificSeminarsShape`. The second property concerns the value of the time to attend the seminars (`unige:timeToAttend`) must have. This constraint declares that the attendance at the seminars is scheduled for `unige:_5_days`. The third property deals with `unige:deadlineToAttend` whose values must have datatype

`xsd:dateTime` and establishes the conditions that the deadline to attend seminars must be less than the `unige:defenceDate`. Therefore, the PhD student has to attend advanced scientific seminars before the date of the thesis defence.

```

unige:PhDStudentShape
  rdf:type sh:NodeShape ;
  sh:property [
    sh:path unige:attends ;
    sh:node unige:AdvancedScientificSeminarsShape ;
  ] ;
  sh:property [
    sh:path unige:deadlineToAttend ;
    sh:datatype xsd:date ;
    sh:lessThan unige:defenceDate ;
  ] ;
  sh:property [
    sh:path unige:defenceDate ;
    sh:datatype xsd:date ;
  ] ;
  sh:property [
    sh:path unige:timeToAttend ;
    sh:hasValue unige:_5_days ;
  ] ;
  sh:targetClass unige:PhDStudent ;
.

```

Listing 3: SHACL-based Rule Advanced scientific seminars

Figure 8: The instantiation of SHACL-based Rule Advanced scientific seminar (pass and fail validation)

We set the instance `unige:Jennifer`, who is a PhD student. Figure 8 (left side) shows the following outcomes: `unige:Jennifer` attends the (`unige:MachineLearningSeminar`) for 5 days. Her last day of attendance is on

December 20th, 2022, which is seven days before the date of the thesis defence (December 27th, 2022). The rule passes the validation.

In case we set different data for the instance `unige:Jennifer`, we have different outcomes as shown in Figure 8 (right side). We note one error message concerning `unige:defenceDate` because the value of this property (December 27th, 2022) is less than the deadline to attend a seminar (December 27th, 2022). This is not allowed since the rule requires that a PhD student has to attend advanced scientific seminars before the date of the thesis defence. The rule fails the validation.

8 Role-centric views

As mentioned above, we intend to provide a role-centric view for each stakeholder, allowing coordinated role-based actions. Actions trigger subsequent requests or changes in other roles' views. In the co-design process, a first series of mockups for PhD student roles, PhD study advisor and for the PhD program director roles have been designed with the Figma³ service. After validation with the stakeholders, we developed an actual implementation of the role-centric views. The various views are automatically synthesised from an XML file capturing the PhD student regulations, i.e. among others the diverse rules discussed above. These views capture the various tasks and deadlines to perform along the PhD. The different actors can interact through the service by performing actions specific to their roles, for instance, a PhD student can submit her thesis subject, which the study advisor validates afterwards. The service updates accordingly the monitoring line of the student.

Figure 9 shows the PhD program study advisor view. It proposes both the status of the admission's list at a glance (here two students) and the monitoring line of all enrolled students. Figure 9 shows also the details of the current status of one of the students. The current implementation provides different views and for each view different levels of granularity (from the general timeline to a specific task), it supports role-based coordinated actions, and allows efficient monitoring of the students. Source code, reports, presentations and videos can be retrieved on request from the Unige GitLab repository⁴.

9 Conclusion

Following a service science approach, and as part of a digital innovation process developed at the University of Geneva, we are working on a service for automating study regulations and study plans, providing a series of role-centric views and actions. We focus on the case study of PhD students.

We showed how to express study regulations as a set of rules that follow a generic model that allows expressing the whole variety of cases present in the regulations. The implementation of the rules instantiates the study regulation and study plans, and provides a monitoring service compliant by design.

Future work encompasses different aspects. First, completing the development of the functional prototype described in this paper; in particular, integrating together all the components (semantic reasoning, back-end, database and front-end). Second, we will continue developing the roadmap we defined in a previous ideation effort [2] and focus on automatically extracting the rules from the textual version of the regulation; and provide a service for helping to correctly write the regulations (i.e. without contradictions). Third, over time, we observe that the practice moves away from the regulations. For example, 90% of the students

³<https://www.figma.com>

⁴<https://gitlab.unige.ch/courses1/pt2/2223/g3>, <https://gitlab.unige.ch/hackademia2022>

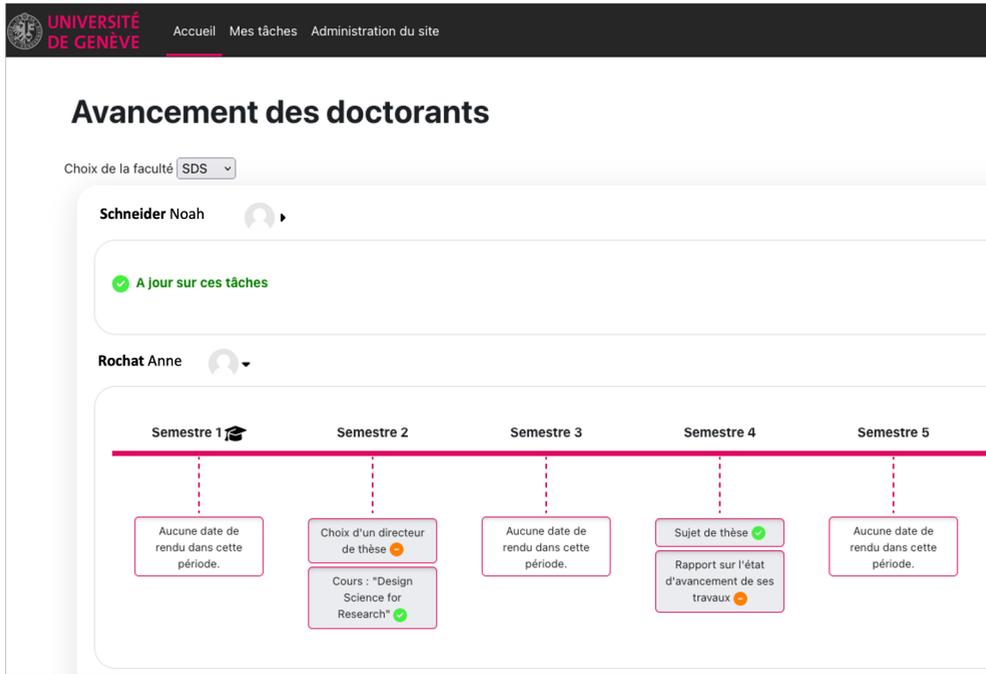


Figure 9: PhD study advisor role - Monitoring of enrolled PhD students

benefit from an extension of the deadline for the submission of the thesis subject. Using AI techniques (e.g machine learning), we determine the gap between the rules and the practice, in order to suggest recommendations (feedback) for changes to the rules in order to integrate new practices and tighten the gap between the regulations and the reality (e.g. recommending a deadline of 3 semesters instead of 2 for submitting the thesis subject). Our objective is to provide a similar service for other programs, in particular BSc and MSc programs. An additional group of students is currently investigating how to scale up the idea and provide a service that can automatise 300 programs.

References

- [1] G. Di Marzo Serugendo, L. Friha, *Digital innovation process in the academic sector - The case of the University of Geneva*, in *International Conference on Exploring Service Science (IESS 2.1)* (ITM Web Conf., 2021), Vol. 38, p. 03004
- [2] G. Di Marzo Serugendo, L. Friha, P.Y. Burgi, J. Döbeli, A. Wade, M.A. Cappelli, A.F. Cutting-Decelle, M. Vonlanthen, N. Durand, A. Hugentobler, *Towards a digital service to help the elaboration, implementation and follow-up of study regulations at the University of Geneva - a hands-on experiment*, ITM Web Conf. **41**, 03004 (2022)
- [3] F. Lautenbacher, B. Bauer, C. Seitz, *Semantic Business Process Modeling - Benefits and Capability*, in *AI Meets Business Rules and Process Management, Papers from the 2008 AAAI Spring Symposium, Technical Report SS-08-01, Stanford, California, USA, March 26-28, 2008* (AAAI, 2008), pp. 71–76
- [4] J.V.F. Dombeu, M. Huisman, *Engineering Semantic Web Services for Government Business Processes Automation*, in *Electronic Government and the Information Systems*

- Perspective - 4th International Conference, EGOVIS 2015, Valencia, Spain, September 1-3, 2015, Proceedings*, edited by A. Ko, E. Francesconi (Springer, 2015), Vol. 9265 of *Lecture Notes in Computer Science*, pp. 40–54, https://doi.org/10.1007/978-3-319-22389-6_4
- [5] M. Born, F. Dörr, I. Weber, *User-Friendly Semantic Annotation in Business Process Modeling*, in *Web Information Systems Engineering - WISE 2007 Workshops, WISE 2007 International Workshops, Nancy, France, December 3, 2007, Proceedings*, edited by M. Weske, M. Hacid, C. Godart (Springer, 2007), Vol. 4832 of *Lecture Notes in Computer Science*, pp. 260–271, https://doi.org/10.1007/978-3-540-77010-7_25
- [6] S. Bala, G. Havur, S. Sperl, S. Steyskal, A. Haselböck, J. Mendling, A. Polleres, *SHAPEworks: A BPMS Extension for Complex Process Management*, in *Proceedings of the BPM Demo Track 2016 Co-located with the 14th International Conference on Business Process Management (BPM 2016), Rio de Janeiro, Brazil, September 21, 2016*, edited by L. Azevedo, C. Cabanillas (CEUR-WS.org, 2016), Vol. 1789 of *CEUR Workshop Proceedings*, pp. 50–55, <http://ceur-ws.org/Vol-1789/bpm-demo-2016-paper10.pdf>
- [7] J.P. Castellanos Ardila, B. Gallina, F. Ul Muram, *Compliance checking of software processes: A systematic literature review*, *Journal of Software: Evolution and Process* **34**, e2440 (2022), <https://onlinelibrary.wiley.com/doi/pdf/10.1002/smr.2440>
- [8] M.E. Kharbili, S. Stein, *Policy-Based Semantic Compliance Checking for Business Process Management*, in *Proceedings of the Workshops colocated with the MobIS2008 Conference: Including EPK2008, KobAS2008 and ModKollGP2008, Saarbrücken, Germany, November 27-28, 2008*, edited by P. Loos, M. Nüttgens, K. Turowski, D. Werth (CEUR-WS.org, 2008), Vol. 420 of *CEUR Workshop Proceedings*, pp. 178–192, <http://ceur-ws.org/Vol-420/paper13.pdf>
- [9] K. Sapkota, A. Aldea, D.A. Duce, M. Younas, R. Bañares Alcántara, *Towards Semantic Methodologies for Automatic Regulatory Compliance Support*, in *Proceedings of the 4th Workshop on Workshop for Ph.D. Students in Information & Knowledge Management* (Association for Computing Machinery, New York, NY, USA, 2011), PIKM '11, p. 83–86, ISBN 9781450309530, <https://doi.org/10.1145/2065003.2065021>
- [10] T.H. Beach, Y. Rezgui, H. Li, T. Kasim, *A rule-based semantic approach for automated regulatory compliance in the construction sector*, *Expert Systems with Applications* **42**, 5219 (2015)
- [11] N. Lohmann, *Compliance by design for artifact-centric business processes*, *Information Systems* **38**, 606 (2013), special section on BPM 2011 conference
- [12] P. Maglio, J. Spohrer, *Fundamentals of service science*, *J. of the Acad. Mark. Sci.* **36**, 18 (2008)