Human-Centered Ontology Engineering in E-Government Environments

Richard Siegl, Bernd Stadlhofer, and Peter Salhofer
FH JOANNEUM
University of Applied Sciences, Graz
emails: {richard.siegl.aim11@fh-joanneum.at, bernd.stadlhofer@fh-joanneum.at, peter.salhofer@fh-joanneum.at}

***Abstract*—Ontology engineering is a relatively new field in computer and information sciences. Its primary goal is to develop methodologies for modelling and building ontologies. These ontologies represent knowledge as a set of concepts within a specific domain. A common problem is, though, that it is almost impossible for domain experts to design and model their own ontology in the domain of E-Government without having the basic knowledge of computer science, especially in the field of ontology engineering. The goal of this paper is to describe, how the Rich Ontology Creation Kit for E-Government Transition (ROCKET), an ontology creation tool based on the Eclipse RCP framework, supports legal experts to bridge the gap between domain and technical specialists. To accomplish that goal, the web application SeGoF is described, which uses ontologies as an input for the automatic generation of E-Government forms based on semantic descriptions. Moreover, the methodology "Ontology Driven E-Government" is explained, as well as, the applied human-centered approach in ROCKET. As a result, the tool ROCKET allows domain experts to design and model ontologies in a convenient way by hiding the ontology language. It basically offers an easy-to-use graphical user interface, including wizards and editors like the axiom modeler to create the necessary elements to model an ontology.**

Keywords-SeGoF; ROCKET; ontology; engineering; ontology modelling; semantic MDA; e-government; domain modelling.

I. INTRODUCTION

Basically, ontology engineering is a relatively new field in computer and information science. Its goals are to develop methods and methodologies for building ontologies which formally represent knowledge as a set of concepts within a specific domain and its relationship to other concepts. These ontologies can be very useful in different situations, for example, when designing semantic web applications, like SeGoF [2].

However, the problem is that it is nearly impossible for users to formally express an ontology for a specific domain without having basic knowledge of computer science, especially in the field of ontology engineering. Therefore, tools and methods are required to help users to engineer their own ontology. Basically, there are at least two different roles which are involved in the process of creating ontologies: one domain expert and one technical expert. The domain expert has the required knowledge of a specific field and the technical expert understands the required concepts of ontology engineering. Both roles are interdependent and essential to create and model ontologies.

This paper focuses on the field of E-Government, more precisely, on the SeGoF web application and its requirements concerning ontology engineering. Therefore, SeGoF is described, as well as the different approaches like semantic MDA and ODEG. As a next step, the human-centered ontology engineering approach will be explained, covering the motivation behind this approach and other available methodologies. The next section covers the ODEG modelling process including all relevant process steps. In the following and last section of this paper, some of the most important features of ROCKET will be presented and explained, how they work.

II. SEMANTIC EGOVERNMENT FORMS (SEGOF)

Classical E-Government solutions are mostly aligned along conventional administrative processes. Hereby, a paper-driven workflow of these processes is the normal way to reflect these tasks. The paper-based application form is than simply replaced by an electronic web-form. As a consequence, the process behind these electronic web-forms is very important because it influences the usability and acceptance of the solution, but the prospects of those classical E-Government web-forms are often very limited. Furthermore, different administrative procedures are usually characterized by a set of rules and constraints which have to be met to make a user eligible for a specific service. In classic E-Government environments, the knowledge about these rules is mostly expressed programmatically but not semantically, which is a limiting factor.

A. *Semantic MDA for e-government*

As mentioned by Salhofer and Stadlhofer [1], this specific approach applies the basic principles of MDA (Model Driven Architecture) [1] in combination with a semantic model to the creation of E-Government services. The goal of this intention is that these services will then be available as semantic web services which can be searched for meeting the current goal or desire of a citizen. To be successful, it is important that all required artifacts are automatically generated from the model and that there is no need for manual coding. This enables very short development cycles and very fast adoption to shifting requirements, which is important because public agencies are facing significant budget cuts. As a consequence, resources have to be used as effective and efficient as possible [1].

B. ODEG

The goal of Ontology Driven E-Government (ODEG) [2] is to model ontologies for the E-Government domain which act as a basis for an integrated E-Government environment. The most important point is that these ontologies can be used to assist a citizen by formally expressing a goal that can be used for service discovery and it can also be used to express the necessary input of services which will be used for the subsequent form generation. The collected input data is transformed into a common data interchange standard format, and then, forwarded to the service oriented architecture (SOA) backend which executes the business process [2]. Furthermore, ODEG can be seen as an ontology engineering methodology because its main goal is to model ontologies for the E-Government domain by proving a human-centered approach.

As mentioned by Stadlhofer et al. [2], every relevant public service is semantically modeled and contains references to the required input elements. Moreover, every constraint on the service input element can be expressed by different semantic rules and can be evaluated by semantic reasoners. This allows an automatic creation of forms, for example web forms, and also plausibility checks of data gathered directly from the user. With this approach, the entire logic is now consistently kept in the semantic model. Additionally, another advantage of this approach is that the knowledge of public services becomes available in a machine processable form which allows more functionality than only form creation [2].

III. Human-centered ontology engineering approach

Nowadays, it is wildly argued that ontologies are the primary key to success concerning the realization of the semantic web. As a result, the ontology engineering process plays an important factor when designing and modelling ontologies. Thus, different methodologies were created to support and to document this process. This subchapter examines different approaches and explains the most important facets of the respective methodology. Basically, the focus lies on the investigation of the Ontology Driven E-Government (ODEG) methodology because ROCKET supports the ODEG modelling process and a human-centered approach by empowering legal experts. First of all, it is important to describe the motivation and requirements behind these methodologies.

A. Motivation

This section describes the motivation and the requirements concerning the E-Government domain derived from the SeGoF platform. These requirements refer to the ontology engineering methodology which is mandatory to achieve a well-defined and structured ontology building process and to support domain experts (or legal experts in case of the E-Government domain), so that they can design, model, and build ontologies autonomously without help from technical experts.

1) Domain expert centered approach: The methodology basically has to offer a domain expert centered approach which is the basic top-level goal. The intention of this approach is to empower legal experts so that they can design, model and build ontologies themselves. To realize this intention, the methodology has to support this approach concerning the respective process and by offering tool support which guides users through the building process.

2) Developed for electronic service provisioning in public administration: First of all, the application area of the ontology engineering methodology is important. In case of the SeGoF platform, the domain of E-Government is the respective target area. In general, it should be designed for electronic service provisioning in the context of public administration.

3) Clear and consistent process: The desired ontology engineering methodology should offer a clear and consistent process which guides users through the ontology building procedure. Usually, the process refers to the ontology building life cycle and consists of different phases. Besides the ontology modelling process, it may be important to cover the resulting project management of the project. This for example includes specification phases which covers the possibilities of knowledge retrieval.

4) Tool support: To realize the domain expert centered approach, the desired methodology should offer tools which support users during the ontology creation process. Basically, the tool has to hide the used ontology language behind a graphical user interface. With the help of certain templates, users are guided through the ontology building process. In the best case, legal experts should be able to implement and test the modelled ontology immediately.

5) Consideration of a possible collaborative construction: An essential aspect in the context of public administration is the consideration of legal certainty. In that case, legislation and enforcement of law on all governmental levels has to be ensured. As a consequence, this requires collaboration with a variety of different legal experts. That implies that the ontology creation tool should support and process a collaborative approach, which allows legal experts to collaboratively model, implement, verify and maintain ontologies.

B. Methodologies in the context of public administration

Concerning this paper, the target field in which respective methodologies have to be applied is public administration, more precisely E-Government and electronic public service provisioning. So the question is, which of the listed methodologies is capable of meeting the requirements. To answer this question, Stadlhofer et al. [3] conducted an analysis on different ontology engineering methodologies to investigate, if any available methods support a domain expert centered approach in the context of electronic service provisioning in public administration.

Considering the resulting requirements of the E-Government domain identified in the beginning of this chapter, Stadlhofer et al. came to the conclusion that none of the analyzed methodologies is fully capable to serve as domain expert centered ontology engineering methodology in the context of PA. As a result, only one of the tested methods named “Integrated Modeling Methodology” technically addresses all methodological requirements in an acceptable way, but this method was developed for a different domain, namely organizational learning, which means that it’s direct exploitation for the E-Government domain is very difficult. But, the authors of the methodological comparison analysis mention that aspects and general guidelines of this and other methods can contribute to a future methodology in this field [3].

IV. ODEG MODELLING PROCESS

Basically, there are different roles involved in the ontology engineering process. At least, there are two different individuals, namely domain and technical experts. The focus lies on these two roles although there could be more people involved, for example knowledge engineers or ontology users.

1) Domain expert: Domain experts or, in case of the E-Government domain, legal experts possess the required knowledge about the targeting problem domain. The goal is to extract this knowledge, abstract it and transform it into an ontology. This role is necessary and acts as a key role in the ontology engineering process.

2) Technical expert: Technical experts have the required skills to model, design, verify and implement ontologies. Mostly, they are experts in developing software because ontology engineering is a software engineering-oriented process. Furthermore, technical experts possess knowledge about a specific ontology modelling language like WSML or OWL which enables them to build and model ontologies.

The problem here is that both described roles are mandatory to succeed in building usable ontologies. Basically, there are different solutions available to overcome this particular problem. One possibility could be to convert technical experts into domain experts. But, sadly, this intention is impossible to realize. The same applies to the intention to convert domain experts into technical experts: The effort simply would be too expensive and inefficient which leads to the conclusion that another solution is needed to overcome the primary problem.

Therefore, adequate tools are required to bridge the gap between technical and domain experts. The primary goal here is to hide the used ontology language which enables domain experts to use these tools. Furthermore, an easy to use graphical user interface has to be used which supports users on the process level to ease the general ontology engineering process.

ROCKET [4] represents a pilot project to meet the mentioned requirements. This ontology creation tool focuses on enabling domain experts of the Public Administration domain to autonomously develop, build and implement ontologies without the help of technical experts. As a result, these developed E-Government services should be directly executed by the SeGoF web application.

A. ODEG modelling process

Basically, the ODEG modelling process is divided into different phases. Figure 1 gives an overview of these phases and describes them in more detail. It is important to mention, that these process steps are not strictly aligned. As depicted by the life cycle, different process steps can be performed individually. As shown in the middle of the figure, maintenance is also a relevant step which implies that certain actions have to be repeated to achieve acceptable, correct and useful results concerning the created ontology. Maintenance also implies that it may be important to extend the expressiveness of the actual domain, for example, by creating additional sub concepts and respective axioms, or in other words, to refine and extend the modelled domain. In addition, it is important to mention that this life cycle is only a recommendation for further studies which explicitly have to investigate the ODEG modelling process.

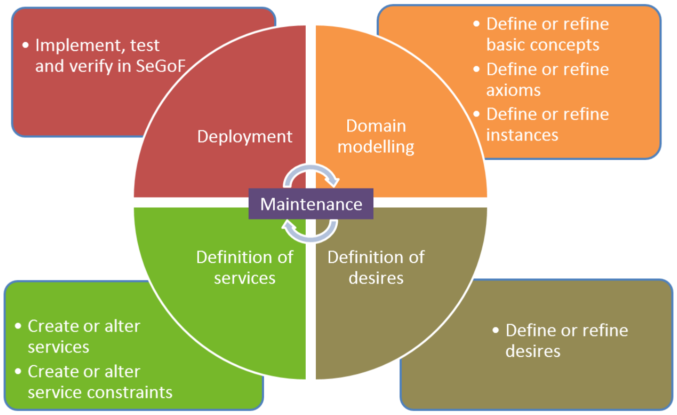

• Implement, test and verify in SeGoF
Deployment
Domain modelling
• Define or refine basic concepts
• Define or refine axioms
• Define or refine instances
Maintenance
Definition of services
Definition of desires
• Create or alter services
• Create or alter service constraints
• Define or refine desires

Figure 1: ODEG modelling process

1) Domain modelling: As mentioned by Stadlhofer et al. [3], service discovery from a citizens point of view should start with selecting the appropriate life situation which are expressed in ODEG by the concept PADomain. For every created domain in the system, a new instance of this concept has be to be defined. The created instance is located in the Domains.wsml file which can be found in the segof/domains folder. Moreover, it is possible to define a list of additional general-purpose ontologies which are relevant for the respective domain [3].

As a next step, it is necessary to model the new created domain. This implies the definition of new concepts, axioms and instances which express the abstracted domain of interest. The following components can be found in the WSML file located in the domain folder in the respective PADomain folder hierarchy.

Definition of concepts: The basic components of every ontology to describe the targeting domain are concepts. Depending on the desired level of detail concerning a domain of interest, several concepts may exist.

Definition of axioms: Furthermore, it may be important to refine the created concepts or to create specific constraints. Therefore, Axioms have to be created which are based on logic programming rules. Of course, it is possible that several axioms may exist in the domain.

Definition of instances: Additionally, it may be relevant to create instances of certain concepts. These instances are also located in the domain WSML file and include concrete values of certain attributes.

2) Definition of desires: After selecting the appropriate life situation, a citizen has to choose one concrete desire or goal he or she wants to reach. For every existing PA domain, it is possible that several desires exist. The isRelatedToConcept attribute expresses the relationship to one or more concepts of the domain and marks the entry point in the ontology.

As mentioned by Stadlhofer et al. [3], in the resulting user interface the user would have the possibility to replace the related concepts in the desire definition by concrete ones. After defining such a concrete desire, an appropriate public service can be discovered for this specific situation.

3) Definition of services: Another important step of the ODEG modelling process is to define specific public web services. Therefore, every semantic service description is split into two different parts. This includes an instance of the geaGrazConstraintPublicService and a corresponding Web Service Modelling Ontology (WSMO) web service. The GEA part includes the description of the relationship to one or more desires and specific constraints that have to be met. By contrast, the WSMO part describes the required input and preconditions [3].

Furthermore, it is possible to create specific constraints which have to be met.

4) Deployment: As the final step in the ODEG modelling process, it is important to implement, test and verify the modelled domain in the SeGoF web application. Therefore, the modelled domain has to be implemented into the respective folder of the SeGoF application, as well as specific entries have to be added to register the new domain. After verifying the created domain, it may be important to refine or extend the existing domain.

V. Human-centered Ontology Engineering with ROCKET

Basically, ROCKET offers several features concerning the modelling of ontologies. The most important characteristics are going to be described in the following chapter. For more information on ROCKET, please see [4].

A. Axiom editor (sub-concept by constraint)

Creating different concepts which have the same super concept is a common step. That implies that the reasoner has to decide either to take the right or the left path through the ontology. Therefore, axioms are required to accomplish that goal and to tell the reasoner which path to choose for a specific situation. In this concrete example depicted in Figure 2, the super concept "PossiblePupil" holds two attributes, namely "age" and "name". If a person is 14 years old or younger, this person will become a "RequiredSchoolPupil". If the person is 15 years old or older, this person will become a "NotRequiredSchool". It is important to mention that these conditions are implemented in respective axioms. Basically, the described scenario is very common, which implies that it would make sense to combine the action of creating concepts and axioms in one step.

Basically, the primary goal of the new "Sub concept by constraint" feature is that users have the possibility to create a new sub concept of an existing super concept and additionally are able to create a respective axiom. Depending on the selected super concept, the available attributes from the super concept should be visible which can be used to create reasonable constraints.

There are two entries in the "Add Entity" context menu in the WSML visualizer when selecting a concept node: the "Sub Concept by Constraint" and "Axiom" option. Selecting the first options allows users to create a sub concept while creating a respective axiom at the same time. As mentioned before, axioms allow to refine concepts and to add specific constraints. Selecting the "Axiom" option allows users, as the name already indicates, to create and model an axiom only. Furthermore, when selecting the root ontology node, the available options differentiate compared to the options. Doing so allows the user to select the "Concept by Constraint" option which creates a normal concept and an additional axiom. In fact, the resulting dialog remains the same.

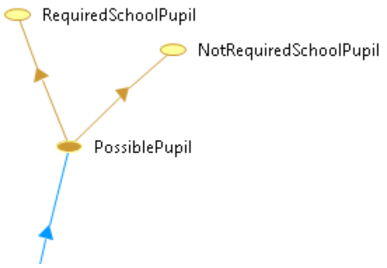
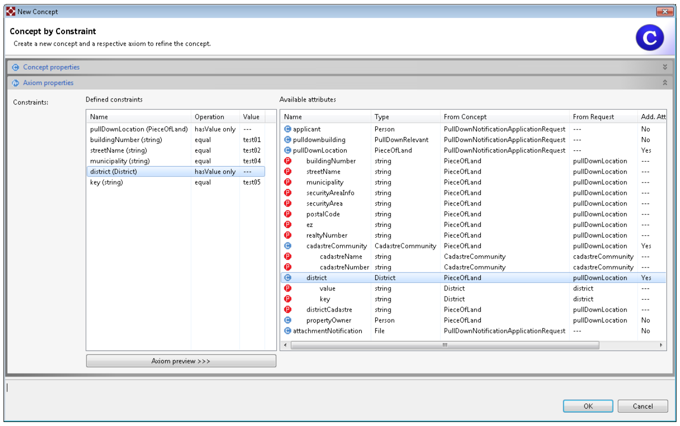


Figure 2: Two different concepts and their super concept



Concept by Constraint

Create a new concept and a respective axiom to refine the concept.

Constraints:

Defined constraints

Name	Operation	Value
pullDownLocation (PieceOfLand)	hasValue only	---
buildingNumber (string)	equal	test01
streetName (string)	equal	test02
municipality (string)	equal	test04
district (District)	hasValue only	---
key (string)	equal	test05

Available attributes

Name	Type	From Concept	From Request	Add. Att
applicant	Person	PullDownNotificationApplicationRequest	---	No
pulldownbuilding	PullDownRelevant	PullDownNotificationApplicationRequest	---	No
pullDownLocation	PieceOfLand	PullDownNotificationApplicationRequest	---	Yes
buildingNumber	string	PieceOfLand	pullDownLocation	---
streetName	string	PieceOfLand	pullDownLocation	---
municipality	string	PieceOfLand	pullDownLocation	---
securityAreaInfo	string	PieceOfLand	pullDownLocation	---
securityArea	string	PieceOfLand	pullDownLocation	---
postalCode	string	PieceOfLand	pullDownLocation	---
ez	string	PieceOfLand	pullDownLocation	---
realtyNumber	string	PieceOfLand	pullDownLocation	---
cadastreCommunity	CadastreCommunity	PieceOfLand	pullDownLocation	Yes
cadastreName	string	CadastreCommunity	cadastreCommunity	---
cadastreNumber	string	CadastreCommunity	cadastreCommunity	---
district	District	PieceOfLand	pullDownLocation	Yes
value	string	District	district	---
key	string	District	district	---
districtCadastre	string	PieceOfLand	pullDownLocation	---
propertyOwner	Person	PieceOfLand	pullDownLocation	No
attachmentNotification	File	PullDownNotificationApplicationRequest	---	No

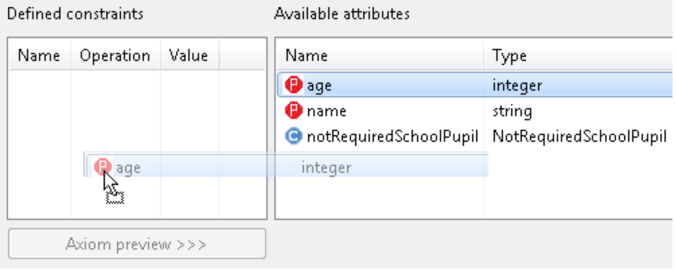
Figure 3: "Sub concept by constraint" dialog

Figure 3 shows the implemented dialog. Basically, the dialog includes basic components like in the normal "New Concept" dialog. It is important to mention that the dialog uses the full size of the display. That means that depending on the size of the ROCKET program, the dialog scales itself to guarantee a better user experience. Furthermore, it is possible to hide all elements which are related to the concept properties. Therefore, users have the possibility to click on the top bar to extend or to hide related user interface elements.

First of all, users have to type in a valid identifier or name for the new sub concept. This name is also used for the name of the respective axiom. Concerning this name, the identifier will be modified by adding the string "Definition". For example, when users want to create a sub concept named "PossiblePupil", the respective axiom name would be "PossiblePupilDefinition". It is also possible to assign specific internationalized human readable descriptions which are important for the SeGoF web application.

Like in the normal “New Concept” dialog, it is possible to add certain annotations and super concepts which influence the resulting concept. Additionally, it is possible to select the “Add relevant imported ontologies” options.

Basically, an axiom consists of different constraints which can be modelled with the help of different gestures. This includes for example the use of drag and drop gestures as demonstrated in Figure 4. Alternatively, another option is to double click on the respective attribute.



Defined constraints

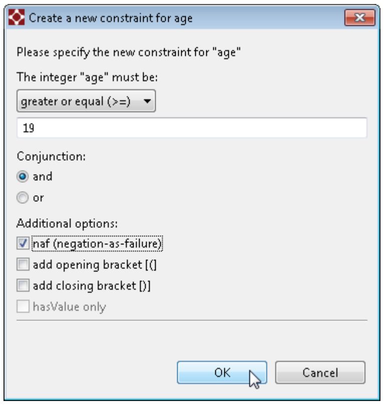
Name	Operation	Value

Available attributes

Name	Type
age	integer
name	string
notRequiredSchoolPupil	NotRequiredSchoolPupil

Figure 4: Creating a constraint while using the drag and drop gesture

As shown in Figure 4, there are different icons displayed for available attributes. The red “P” indicates that the attribute is a primitive data type, while the blue “C” indicates that the respective attribute is another concept which possibly holds additional attributes. To browse through these attributes of other concepts, users have the possibility explore them by double clicking on a concept. As a consequence, additional attributes will be added under the respective concept.



Create a new constraint for age

Please specify the new constraint for "age"

The integer "age" must be:

greater or equal (>=)

19

Conjunction:

● and

○ or

Additional options:

☑ naf (negation-as-failure)

□ add opening bracket [(]

□ add closing bracket [)]

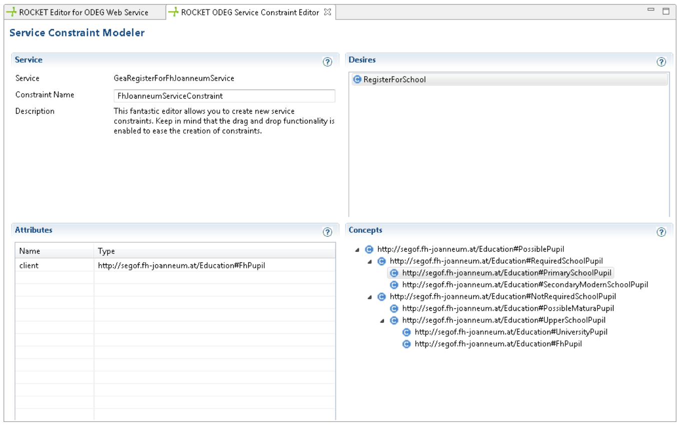
□ hasValue only

Figure 5: Dialog for creating a constraint for the type integer

It is important to mention that depending on the selected type of the attribute, a specific dialog will show up. As shown in Figure 5, the new dialog offers the possibility of adding additional options to the new constraint. In case of an attribute of type integer, the user can choose between different operations like greater, smaller, greater or equal, etc. Of course, a specific number can be added which fulfils the constraint. Furthermore, the desired conjunction can be chosen. In the additional options, users have the possibility of adding the negation keyword “naf”, adding opening and closing brackets and selecting the option “hasValue only”.

B. *Service constraint modeller*

The ROCKET editor for ODEG web service focuses on the management of available services in the domain. On the left side, all available services will be displayed. By selecting another service, all respective parameters can be checked and altered. This includes available service constraints, related desires, service implementations, related service requests and the name of the service. Furthermore, the user can determine if the service should be an information service, which means that as a result, the SeGoF web application provides a document (e.g., a .pdf file) which includes valuable information for the citizen.



ROCKET Editor for ODEG Web Service
ROCKET ODEG Service Constraint Editor

Service Constraint Modeler

Service

Service: GeaRegisterForFhJoanneumService

Constraint Name: FhJoanneumServiceConstraint

Description: This fantastic editor allows you to create new service constraints. Keep in mind that the drag and drop functionality is enabled to ease the creation of constraints.

Desires

RegisterForSchool

Attributes

Name	Type
client	http://segof.fh-joanneum.at/Education#FhPupil

Concepts

- http://segof.fh-joanneum.at/Education#PossiblePupil
    - http://segof.fh-joanneum.at/Education#RequiredSchoolPupil
        - http://segof.fh-joanneum.at/Education#PrimarySchoolPupil
        - http://segof.fh-joanneum.at/Education#SecondaryModernSchoolPupil
    - http://segof.fh-joanneum.at/Education#NotRequiredSchoolPupil
        - http://segof.fh-joanneum.at/Education#PossibleMaturaPupil
        - http://segof.fh-joanneum.at/Education#UpperSchoolPupil
            - http://segof.fh-joanneum.at/Education#UniversityPupil
            - http://segof.fh-joanneum.at/Education#FhPupil

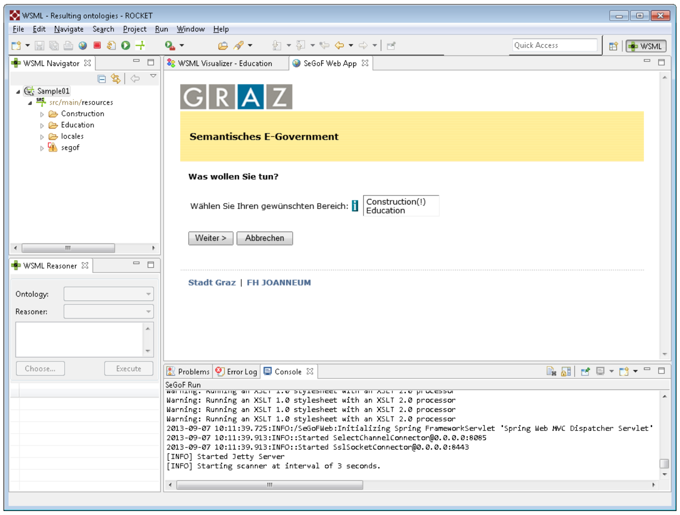
Figure 6: Service constraint modeller

Another important part of the web service editor is the ODEG service constraint modelling editor which pops up by selecting the edit or create button after right clicking into the area of the service constraints. As depicted in Figure 6, users have the possibility to create specific constraints for the respective service. Depending on the selected desire, different concepts will be displayed as a tree. By using intuitive drag and drop gestures, it is possible to select a desired concept and drop it in the attributes area to create an additional attribute. Therefore, the user has to provide a new name. After creating a new or editing an existing constraint, users have to save the changes by pressing CTRL + S. In the background, the new or changed service constraint will be stored in the respective domains file as an instance of type geaGraz#ServiceConstraint including the created attributes.

C. *Automatic implementation of ontologies in SeGoF*

Another feature is called “run button”. As a developed use case confirmed, the test and verification steps of the ontology in the SeGoF web application are rather complicated and not supported by the tool ROCKET. As a result, users have to manually add the generated ontology and the segof wsml files in the respective folder in SeGoF, build the SeGoFOntologies project and start the jetty web server via the console. Since this is not a human-centered approach, additional features have to be added to give users the possibility of implementing, testing and verifying the created ontology directly in ROCKET. Therefore, a new button has to be implemented in the tool which executes the described steps of deploying the ontology in SeGoF automatically. Besides reducing the level of difficulty of the steps which have to be executed manually, users would have the possibility of verifying the ontology very quickly.

1) Main run button: The main run button located on the right side of the four new buttons acts as the primary button to initiate the start of the SeGoF web application. This button has to be executed before using the others. First of all, when pressing this button, ROCKET copies the required SeGoF run files from its sources into the actual workspace. This includes a folder holding the maven binaries, the SeGoFOntologies project and the SeGoFWeb project and other relevant projects from the SeGoF web application, respectively in separate folders. After copying the new folders, the respective WSML files (depending on the selected project in ROCKET) will be implemented in the SeGoFOntologies project and the preconfigured and pre-built SeGoFWeb.war and other required files will be installed in the local maven repository. As a next step, this project will be built using the implemented maven sources. Depending on the operating system of the host system of ROCKET, a specific maven run file and a proper command will be chosen to accomplish this task. That implies that maven is not a prerequisite when users are executing the run button, which is a big advantage. Furthermore, the integrated jetty web server will be started and the internal browser of ROCKET will be opened displaying the correct page with the address “localhost:8080/SeGoFWeb” as shown in Figure 7. Now, users have the possibility of verifying and testing the modelled ontology instantly.



Semantisches E-Government

Was wollen Sie tun?

Wählen Sie Ihren gewünschten Bereich: Construction(!)
Education

Weiter > Abbrechen

Stadt Graz | FH JOANNEUM

```
Warning: Running an XSLT 1.0 stylesheet with an XSLT 2.0 processor
Warning: Running an XSLT 1.0 stylesheet with an XSLT 2.0 processor
Warning: Running an XSLT 1.0 stylesheet with an XSLT 2.0 processor
Warning: Running an XSLT 1.0 stylesheet with an XSLT 2.0 processor
2013-09-07 10:11:39.725:INFO:/SeGoFWeb:Initializing Spring FrameworkServlet 'Spring Web MVC Dispatcher Servlet'
2013-09-07 10:11:39.913:INFO::Started SelectChannelConnector@0.0.0.0:8085
2013-09-07 10:11:39.913:INFO::Started SslSocketConnector@0.0.0.0:8443
[INFO] Started Jetty Server
[INFO] Starting scanner at interval of 3 seconds.
```

Figure 7: Internal browser in ROCKET

2) Refresh button: After executing the main run button for the first time, the jetty web server is up and running. When users modify a respective ontology, they have the possibility to execute the refresh button so that only the new WSML files get copied and the project SeGoFOntologies gets built. The running jetty web server detects the modified jar file from the SeGoFOntologies projects and restarts itself automatically. Of course, it is possible to re-execute the main run button, but the advantage of the refresh button is that it finishes much quicker.

3) Stop button: As the name of the button already indicates, the stop button stops the running jetty web server and closes the internal browser view automatically. This, for example, is useful when quitting ROCKET to close all remaining processes. As usual, the status of this process can be checked in the console view.

4) Open browser button: This button allows users to reopen the internal browser view if the Jetty web server is running. If the server is not running, a respective error message will be displayed telling users that SeGoF is not running. Furthermore, this button refreshes the actual address of the browser to the default address. This is useful when testing different PA domains to return to the initial web page.

VI. Conclusion

First of all, the illustration of the ODEG modelling process is an important step toward a well-defined and structured process concerning the ontology building process in ROCKET. While the depicted life cycle is only a recommendation, it can be seen as an important input for further studies in this field. The ontology creation tool ROCKET offers several specific wizards and editors which allow users to build and model a specific ontology in the field of E-Government which can be used by the SeGoF web application as an input to automatically generate E-Government forms to ultimately invoke a certain service. The top-level goal of ROCKET is to empower domain or legal experts so that they are able to build ontologies autonomously. Furthermore, ROCKET offers an easy to use graphical user interface which guides users through the ODEG modelling process by pursuing a human-centered ontology engineering approach. Additionally, tools like the WSML Visualizer help users understand the modelled ontology more easily. To conclude, it is important to mention that future work has to be focused on conducting usability tests to improve the overall usage of the tool. Consequently, the ontology engineering methodology ODEG has to be further developed to achieve a well-defined and documented process. With the help of SeGoF and ROCKET, the efficiency of public offices concerning the offering of public services can be increased significantly. Because of the fact that legal experts would design and model ontologies autonomously which then will be used as an input for the SeGoF web application, public services could be published immediately.

References

[1] Peter Salhofer and Bernd Stadlhofer, “Semantic MDA for E-Government Service Development”, hicss, pp. 2189-2198, 2012 45th Hawaii International Conference on System Sciences, 2012, http://doi.ieeecomputersociety.org/10.1109/HICSS.2012.524, [retrieved: 10, 2013].

[2] Peter Salhofer, Bernd Stadlhofer, and Gerald Tretter, “Ontology Driven E-Government”, in Politics, Democracy and E-Government: Participation and Service Delivery (Reddick, Ch., editor), Information Science Reference, April 2010, pp. 383-401.

[3] Bernd Stadlhofer, Peter Salhofer, and Augustin Durlacher, “An Overview of Ontology Engineering Methodologies in the Context of Public Administration”, p. 7, 2012.

[4] Bernd Stadlhofer, Peter Salhofer, Augustin Durlacher, and Martin Zierler, “ROCKET - Rich Ontology Creation Kit for E-Government Transition”, in Electronic Government and Electronic Participation, Joint Proceedings of Ongoing Research and Projects of IFIP EGOV and IFIP ePart, Kristiansand, Norway, September 3-6, 2012, Trauner Verlag, pp. 106-114.