Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 415 – 419

ISBN 978-83-60810-14-9
ISSN 1896-7094

Integration of governmental services in semantically described processes in the Access-eGov system

Marek Skokan, Tomáš Sabol
Technical University of Košice, Faculty of Economics, Letná 9, 042 00 Košice, Slovakia
Email: {marek.skokan, tomas.sabol}@tuke.sk

Marián Mach
Technical University of Košice, Faculty of Electrical Engineering and Informatics, Letná 9, 042 00 Košice
Email: marian.mach@tuke.sk

Karol Furdík
InterSoft, a.s. Floriánska 19, 040 01 Košice
Email: karol.furdik@intersoft.sk

***Abstract*—This paper describes a "user-centred" approach to integration of services provided by the government in a "traditional" (i.e. face-to-face) or electronic way, applied in the EU R&D project Access-eGov. Individual ("atomic") services are integrated on semantic level (using semantic description of existing services – either traditional or e-services) into a scenario, realization of which leads to a solution of a problem faced by the end users (citizens or businesses) in a given life event (e.g. how to get a building permit, establish company etc.). Benefits for the end users are twofold: firstly they are provided with higher value-added services – a scenario of services (consisting of a series of services, including their dependencies), not just a single services; the services in the scenario are personalised (i.e. adapted to his/her personal data and/or situation). In case some services in the scenario are available electronically, they can be also executed online, increasing thus ultimate benefits to the end user. First prototype of the Access-eGov platform was test and evaluated in three pilot applications in three EU countries.**

I. Introduction

INTEROPERABILITY was recognised as a precondition for the implementation of European eGovernment services in the eEurope Action Plan [1] and is explicitly addressed as one of the four main challenges in the i2010 EU strategy [2]. One of the most promising approaches to the interoperability is the employment of semantic technologies [3], [4]. Main advantage of this approach is the capability to formally describe meaning and context of government services, both traditional (i.e. face-to-face, "paper-based") as well as electronic ones (provided as electronic forms or web services), without necessity to modify the services themselves. The Access-eGov project (www.access-egov.org) builds on the use of semantic technologies with the aim to enable semantic discovery and semantic integration of governmental services into user specific scenarios. Integration of services into (user specific) scenarios (and the interoperability among them) is based on the WSMO technology (www.wsmo.org) used for description of process models by means of concepts defined in a knowledge model (ontology).

Access-eGov is a R&D project funded by the European Commission within the 6 th Framework Programme (FP6), Information Society Technologies (IST) programme. Within the Access-eGov project a SW platform supporting semantic interoperability of traditional as well as electronic government services in practical applications, together with methodological guidelines for introduction and management of such a platform, are being developed. In contrast to other projects, Access-eGov applies rather front office integration approach, i.e. no changes on the back office side are required.

The technological solution developed within the project is tested and evaluated within three pilot applications (in Slovakia, Poland and Germany) and one lab test (Egypt). The pilot application in Slovakia deals with the administration process of obtaining a building permission. The pilot application in Poland deals with the administration process of establishing an enterprise. The pilot application in Germany involves administrative and some non-administrative activities that are necessary to perform in a getting-married scenario.

II. Semantic description of services

A. Web Service Modelling Ontology (WSMO)

The WSMO framework (www.wsmo.org) provides a consistent conceptual model with the inclusion of mediators and distinction between goals and capabilities [9]. The Web Service Modelling Ontology (WSMO) is a conceptual model for describing semantic Web Services. WSMO consists of four major components: ontologies, goals, Web Services and mediators. Ontologies provide formal semantics to the information used by all other components. WSMO specifies the following constituents as part of the description of ontology: concepts, relations, functions, axioms, and instances of concepts and relations, as well as non-functional properties, imported ontologies, and used mediators. The latter allows the interconnection of different ontologies by using mediators that solve terminology mismatches.

A goal specifies objectives that a client might have when consulting a Web Service, i.e. functionalities that a Web Service should provide from the user perspective. In the WSMO, a goal is characterized by a set of non-functional properties, imported ontologies, used mediators, the requested capability and the requested interface.

A Web Service description in WSMO consists of five subcomponents: non-functional properties, imported ontologies, used mediators, a capability and interfaces. The capability of a Web Service defines its functionality in terms of preconditions, postconditions, assumptions and effects. A capability (therefore a Web Service) may be linked to certain goals that are solved by the Web Service via mediators. Preconditions,

assumptions, postconditions and effects are expressed through a set of axioms and a set of shared all-quantified variables. The service interfaces are described in the following chapter.

Mediators describe elements that aim to overcome structural, semantic or conceptual mismatches, which can appear between the different components that build up a WSMO description. Currently the specification covers four different types of mediators:

- OOMediators - import a target ontology into a source ontology by resolving all the representation mismatches between the source and the target;
- GGMediators - connect goals that are in a relation of refinement allowing the definition of sub-goal hierarchies and resolve mismatches between those;
- WGMediators - link a goal to a Web Service via its choreography interface meaning that the Web Service fulfils the goal; or link a Web Service to a goal via its orchestration interface meaning that the Web Service needs this goal to be resolved in order to meet the functionality;
- WWMediators - connect several Web Services for collaboration.

B. WSMO Choreography and Orchestration

Interface of a Web Service provides further information on how the functionality of the Web Service is achieved. It describes the behaviour of the service from the client's point of view (service choreography) and how the overall functionality of the service is achieved in terms of cooperation with the other services (service orchestration).

A choreography description is semantically based on the Abstract State Machines (ASMs) [5] and consists of the states represented by ontology, and the if-then rules that specify (guarded) transitions between states. The ontology that represents the states provides the vocabulary of the transition rules and contains the set of instances that change their values from one state to another. The concepts of an ontology used for representing a state may have specified a grounding mechanism, which binds service description to a concrete message specification (e.g. WSDL).

For the Orchestration interfaces, it is planned by the WSMO authors to proceed as follows. The language will be based (note, that it is envisioned only, and the specification is not finished yet) on the same ASMs model as Choreography interfaces which - in order to link to externally called services or (sub)goals that the service needs to invoke to fulfil its capability - needs to be extended as follows:

- Goals and Services can be used in place of rules, with the intuitive meaning that the respective goal/service is executed in parallel to other rules in the orchestration
- The state signature defined in the choreography can be reused, i.e. external inputs and outputs of the service and the state of the choreography can be dereferenced also in the orchestration
- Additionally the state signature for the orchestration interface can extend the state signature of the choreography interface, with additional in/out/shared/controlled concepts which need to be tied to the used services and rules by mediators
- Respective WW or WG mediators need to be in place to map the in and out concepts defined in the orchestration to the respective out and in concepts of the choreography interfaces in the used services and goals, i.e. these mediators state which output concepts are equivalent to which input of the called service/goal and vice versa

C. Modifications in the Access-eGov project

The life event approach [6] was adopted for modelling of government services, where the life event concept plays a central role – as a formal representation of user's point of view, his/her needs and requirements. Implementation of this approach resulted in the necessity to add the following top-level WSMO elements to the WSMO specification:

- Life Events – as formal models of user's needs, consisting from multiple goals and services organised into a generic scenario and expressed by orchestration construction consisting from shared variables (i.e. instances of concepts that are used within this life event) and transition rules that enable customisation of the generic scenario into a user specific scenario based on the user situation (i.e. instances describing this situation).
- Services as a generalisation of Web service concepts. This approach enables to describe both electronic and traditional government services by means of a service profile, containing functional and non-functional properties, capabilities, and interfaces. If there is no executable service available for a traditional service, the textual description of the required inputs (e.g. documents and forms, etc.) and requested actions (e.g. visit of a particular office) is specified as the non-functional property.

Requirement-driven approach [7] was developed within the Access-eGov project to guide semantic modelling and annotation (i.e. description of services by means of ontological models) of services provided by the government. While goals and life events are modelled in the ontologies (knowledge models) developed within this approach, the result of the annotation is a formalised WSML representation of the ontology containing all the definitions (concepts, classes) of services.

D. Process description in the Access-eGov

The current WSMO specification for the process model based on the ASMs is, based on experience in the Access-eGov project, not structured in a way suitable for interaction with human actors, which is required for eGovernment applications especially those supporting also traditional services. For this reason, we have designed and implemented a workflow-based extension to the WSMO specification. Besides the objectives to guide citizens to achieve specific goals, and to coordinate activities performed by all actors - citizens, traditional public administration services and web services, the following facilities were identified as useful for a process model to provide support for modelling orchestrated scenarios:

- compatibility with the standard process modelling notation (i.e. BPMN) in order to visualize scenarios to users and to use standard tools for modelling;
- compatibility with the proposed standard workflow modelling languages (i.e. WS-BPEL).

The Access-eGov model is based on the workflow CASheW-s model. The state signature is reused from the WSMO specification and replaces the ASMs transition rules with the workflow constructs. Shared ontology state signature allows reusing grounding of the input and output concepts to relevant communication protocols via WSDL for invocation of web services. Workflow model consists of activity nodes connected with the control or dataflow links. Each node can be either an atomic node (Send, Receive, AchieveGoal and InvokeService), or a control node (Decision, Fork and Join).

III. Solution of life event

To put it simply, a WSML representation of a generic scenario is associated to the specified life event. This representation is then interpreted by the Acces-eGov system and presented to the user via SW tool called Personal Assistant client. The user answers relevant questions and if needed s/he chooses from a list of provided services.

The process of solving the life event situation consists of a set of specific goals that should be achieved, as well as from activities performed by all actors - citizens, traditional public administration services and web services. All these aspects are part of the process model (i.e. process ontology comprising generic scenarios) that is the core control (transition rules) and data (shared variables and data mediators) structure of the Access-eGov platform. Thus, process ontologies can be seen as an interface between the technical infrastructure design and the pilot applications. They provide a specification of the inner data structure for system components responsible for discovery, composition, mediation, and execution of services [8].

```
interface MarriageLifeEventInterface↓
  orchestration↓
  sharedVariables { ?inputQ1, ?output, ?ApplyForMarriageOutput }↓
  transitionRules↓
      perform receive ?inputQ1 memberOf Q1.↓
        nfp↓
          aeg#configuration hasValue _boolean("true")↓
        endnfp↓
      perform achieveGoal ApplyForMarriageGoal↓
                      usesMediator ppMediator↓
                          dataFlow↓
                              ?inputQ1 => ?q1.↓
                              ?ApplyForMarriageOutput <= ?ApplyForMarriage.↓

      perform achieveGoal WeddingPlaceReservationGoal↓
                      usesMediator ppMediator↓
                          dataFlow↓
                              ?inputQ1 => ?q1.↓
                              ?output <= ?a1.↓

      perform achieveGoal WeddingCeremonyGoal↓
                      usesMediator ppMediator↓
                          dataFlow↓
                              ?inputQ1 => ?q1.↓
                              ?output <= ?a1.↓
```

Fig. 1 Fragment of the process ontology of the pilot application of marriage in the new Access-eGov WSML notation

The above-presented figure is the high level process description of the life event getting marriage in Germany. The interface (MarriageLifeEventInterface) consists of two parts: sharedVariables and transitionRules. The first part defines variables (i.e. instances of concepts) that are visible within the whole interface of goal. Second part defines the process itself by using constructs from the set of the following constructs: if-then-enfIf, achieveGoal, send, receive. Note, that the example above uses only achieveGoal construct since it is high level process model (kind of complex goal) that is decomposed into three sub-processes (sub-goals).

In the Access-eGov syntax for process description the construct if-then-endIf is branching rule. This rule is used when we need to decide whether some constructs will be executed or not. The decision is done based on the evaluation of condition in the form of logical expression written in WSML syntax. When goal need to be decomposed into subgoals the construct achieveGoal is used to address one of such sub-goal. There are three sub-goals of the presented goal (life event) marriage (ApplyForMarriageGoal, WeddingPlaceReservationGoal, WeddingCeremonyGoal). As can be seen the variables are mediated between goal and its subgoals (construct usesMediator). The operator '=>' means that variable known in goal (that is on the left side of the operator) is known in sub-goal as variable on the right side of the operator (i.e. it's data mediation form goal to sub-goal). The operator '<=' means that variable known in goal (on the left side) holds data from variable known in sub-goal (on the right side) – i.e. data mediation form sub-goal to goal. The construct send means that process sends instance of specific concept to user. The construct receive means that the process needs instance of specific concept from the user side.

A. Presentation and interpretation

The fragment presented above defines a part of the application that is presented to the user via the Personal Assistant Client. A screenshot of the Personal Assistant Client is depicted in Figure 2.

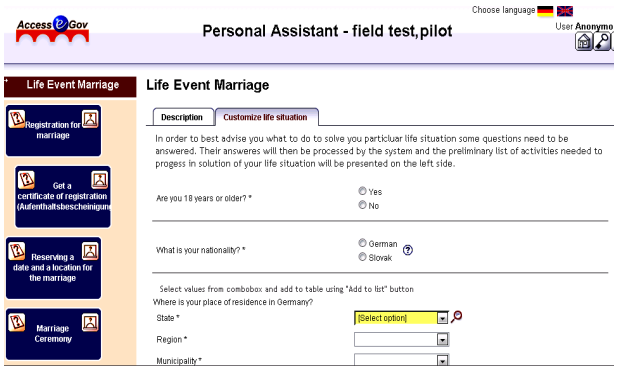

Fig. 2 Fragment of the German pilot application presented to users, that is defined in the presented Access-eGov WSML notation

Customisation of the user situation is based on the answers obtained from the user side that are internally (i.e. in the system) held as values within the instances of the concepts that are used in the process ontology. In this case the instance ? q1 of the concept Q1 holds answers to the questions about age, nationality and place of residence of the user. These an-

swers are then used for the process customisation (i.e. insertion of sub-goal(s) or possibly for withdrawal of sub-goal(s) – not use in the current Access-eGov yet-) as well as for the service filtration.

Note, that on one hand those goals that do not have sub-goals (that cannot be decomposed) are considered as goals that might be resolved with the atomic administration service, or they represent complex part of the process that are not modelled. The latter means for the user that it is not possible to identify specific type of governmental services (and thus there are not instances of such kind of services described semantically in the system). Textual description of such goal is intended for navigation of the user. The example may occur in the German pilot application (getting married) in case that the spouse was born outside the EU and does not have German citizenship. Such case is not very generic and the set of required goals (most likely achievement of specific documents by using specific services) is not modelled. Currently, it's modelled just for cases when spouse is from EU in the Access-eGov system.

On the other hand those goals that contain sub-goals are considered as solved via services that resolve their sub-goals and with services that resolve them. In German pilot application, the example is goal 'Registration for marriage' that contains sub-goal 'Get a certificate of registration'. There is a specific service for both of these goals and the first goal is solved via the use of both of these services.

Note, that the existence of the service(s) to the goal (i.e. the existence of the specific kind of service that might be used by the specific user – e.g. in terms of place of user residence) is known to the user by the picture of office window drawn in the rectangle representing goal. A screenshot of the Personal Assistant Client with identified service is depicted in Figure 3.

Fig. 3 Fragment of the German pilot application presenting service details of the identified service to the goal "Get a birth certificate"

Simply, after matching capabilities of the goal against capabilities of semantically described services the AeG system obtains service(s) that resolve goal. These capabilities are in the form of WSML logical expressions. The overall matching mechanism is not presented in this paper (more information can be found e.g. in [9]).

B. Practical experience with process description

The evaluation of the first Access-eGov prototype was done within the first trial from October 2007 to the end of January 2008. Within this trial also quality of the ontologies and process models was evaluated. The results of this evaluation were analysed and implied changes that are currently being implemented. The second prototype will be tested and evaluated in the second trial in autumn 2008. The second prototype will incorporate also improved (easier) syntax for process description. Comparison of the old and new syntax is provided in Figure 1 (the fragment describes the life event getting married).

```
interface MarriageLifeEventInterface↓
  orchestration↓
    workflow↓
      perform n1_q1 receive ?q1 memberOf Q1.↓
        nfp↓
          aeg#configuration hasValue _boolean("true")↓
        endnfp↓
      perform n1_1g achieveGoal ApplyForMarriageGoal↓
      perform n1_2g achieveGoal WeddingPlaceReservationGoal
      perform n1_3g achieveGoal WeddingCeremonyGoal↓

    controlFlow↓
      source n1_q1 target n1_1g↓
      source n1_1g target n1_2g↓
      source n1_2g target n1_3g↓

    dataFlow↓
      source n1_q1{?q1} target n1_1g{?q1}↓
      source n1_q1{?q1} target n1_2g{?q1}↓
      source n1_q1{?q1} target n1_3g{?q1}↓

      source n1_q1{?q1} target n2_1d{?q1}↓
      source n1_q1{?q1} target n2_fd{?q1}↓
      source n2_1o{?a1} target n1_1g{?a1}↓
      source n1_q1{?q1} target n2_0g{?q1}↓
```

Fig. 4 Fragment of the process ontology of the pilot application of marriage in the old Access-eGov WSML notation

The new syntax significantly simplifies the activity of process description. The most important positive aspect of the new syntax of process description is that it is not necessary to associate all usages of variables to the concrete node since the shared variables are known within the whole goal and the variables mediated between goal and its sub-goals are known within the whole sub-goals. This implies that identifiers of nodes are not needed. Another important aspect is that it is not necessary to define flow among the nodes. The process is read by the Access-eGov core system (execution mechanism) as a sequence as it is natural for the human reader too. Note, that the process description is done in the text based editor.

IV. Conclusions

The Access-eGov system provides a consistent solution for description of processes within public administration, their interpretation and presentation to the user. This paper is focussed on the process description and some results of the first trial evaluation. The formalisms for the process description used in the Access-eGov project, represent an upgrade of the WSMO process description. This upgrade is based on the workflow CASheW-s model, therefore it is considered as compatible with the standard process modelling notation (i.e. BPMN) as well as compatible with the proposed standard workflow modelling languages (i.e. WS-BPEL). The first compatibility enables to visualize scenarios process models

(scenarios) to the users and to use standard tools for modelling. Experiences gained so far within the Access-eGov show that the first version of the formalism proposed for the process description was difficult to read (understand) by public servants. For this reason, the syntax for process description was simplified, what will enable to check the correctness of the process description by public servants (i.e. not IT experts) and also to make corresponding changes (if needed). Thank to this, the administration of the Access-eGov system will be more flexible and easier, and the corresponding overheads lower.

Acknowledgment

The Access-eGov project is co-funded by the European Commission within the contract No. FP6-2004-27020. The work presented in the paper was also supported by the Slovak Grant Agency of the Ministry of Education and Academy of Science of the Slovak Republic within the 1/4074/07 Project "Methods for annotation, search, creation, and accessing knowledge employing metadata for semantic description of knowledge".

References

[1] eEurope 2005: An information society for all. COM (2002) 263 final of 28 May 2002, http://europa.eu.int/information_society/eeurope/2005/ all_about/action_plan/index_en.htm

[2] i2010—A European Information Society for growth and employment, COM (2005) 229 final of 1 June 2005, http://ec.europa.eu/i2010

[3] Jochen Scholl, Ralf Klischewski, E-Government Integration and Interoperability: Framing the Research Agenda, in: *International Journal of Public Administration,* Vol. 30, Issue 8-9, 2007, pp. 889-920.

[4] A. Abecker, A. Sheth, G. Mentzas, L. Stojanovich (eds.), Proceedings of AAAI Spring Symposium "Semantic Web Meets eGovernment" (Stanford University, March 27-29, 2006), Technical Report SS-06-06, AAAI Press, Menlo Park, CA, 2006.

[5] Roman, D., Scicluna, J., Fensel, D., Polleres, A., de Bruijn, J.: D14: Ontology-based choreography of WSMO services. WSMO working draft, DERI (2006) Available online at http://www.wsmo.org/TR/d14/v0.4/ .

[6] Anamarija Leben, Mirko Vintar, Life-Event Approach: Comparison between Countries, in: *Electronic Government,* Springer LNCS 2739, 2003, pp. 434-437

[7] Klischewski, R., Ukena S.: Designing Semantic e-Government Services Driven by user Requirements. In: *Proceedings of ongoing research, project contributions and workshops, 6th International EGOV Conference,* September 3-6, 2007, Regensburg, Germany, pp. 133-140 (2007)

[8] Access-eGov Platform Architecture. Deliverable D3.1, Access-eGov Project, 2006. Available at http://www.accessegov.org/acegov/uploadedFiles/webfiles/cffile_10_17_06_9_41_59_AM.pdf

[9] Marek Skokan, Peter Bednar, Semantic orchestration of services in eGovernment, in: V. Snášel (ed.), *Proc. of Znalosti (Knowledge)* 2008. STU, Bratislava, Slovakia, 2008, pp. 215-223