Towards an Implementation of the EU Services Directive with Semantic Web Services

Ingo Weber and York Sure

SAP Research, CEC Karlsruhe, Germany
{ingo.weber,york.sure}@sap.com

Abstract. The EU Services Directive aims at easing the burdens for the EU's citizens to open up new businesses by providing a single-point-of-contact for the complete business lifecycle. It has to be implemented by all EU member states by the end of 2009. The key technical challenge is the strong dependency between the individual situation of each citizen and the underlying business process: the resulting large variability of possible processes makes it difficult to pre-configure a system for dealing with all required variants. To overcome this bottleneck we present a method and a prototype for automatic service composition, visualization, monitoring, and execution based on Semantic Web Services.

1 Introduction

1.1 Motivating Scenario

Assume that Pablo Benitez, a Spanish citizen, wants to move to the German city Cologne to open up a café with the possibility to sell liquor. In order to get permission to open the café, he has to register as a resident in Cologne; obtain proofs of his record from the police, the Registry of Trade and Industrial Offences, and the tax office; register with the commercial registry; obtain a tax number and tax clearance for the new business; get the liquor license from the office of business affairs; and finally register his new business. Many of these steps require him to fill in a number of forms, giving away – among other information – his current, new, and business addresses, information about his marriage, both of his parents, whether he participated or signed up for an integration course, and so on. Most of the forms today are only available in German; and while his German language skills may be sufficient to operate a café and to interact with customers, the language used in public administration may use an entirely different vocabulary. Further, he is likely to fill in the same data into many different forms, e.g., his addresses. Eventually, the forms need to be signed and delivered to the various offices, sometimes he has to show up personally. Finding out the steps and the dependencies of the business process in first place is a difficulty in itself as each opening of a business is a highly individual process that is dependent on the individual situation of the founder, the kind of business and the location of the business. Clearly, these circumstances make Pablo think twice, before making the step to open up the café; and may even make him regret this decision various times before the actual opening.

W. Abramowicz (Ed.): BIS 2009, LNBIP 21, pp. 217–227, 2009.

1.2 The EU Services Directive and Related Work

Having understood these hardships and the negative economic aspects and hindrance for a more tightly integrated Europe, the European Union decided to improve the situation by means of the so-called "EU Services Directive" [1]. This directive requires all public entities at the level of a city council or "Gemeinde" to offer their services, at least in part, over the internet. In the above example, Pablo has to have the opportunity to request the licenses irrespective of his current location, while being able to monitor the progress of the licensing process remotely.

Key requirement of the EU Services Directive is to establish a Single-Point-of-Contact (SPoC) that serves as the sole interface to the citizen and coordinates the execution of the application process – and later on for the complete lifecycle of a business. The law does not prescribe how such a SPoC has to be provided, be it via internet or via a call center, nor on which granularity it has to be established, may it e.g. be on a federal state level or on a city level. These issues are still under discussion in the EU member states.

For this paper we assume the implementation as a Web-based solution. The level of granularity (federal state or city or …) can in principle be neglected, however, we assume that there will be exactly one or a few SPoCs for e.g. each of the German federal states. We further assume that SPoC interacts with the applicant via the Web and provides a Web-based questionnaire as the user interface to the applicant in which all necessary data for handling the licensing request is collected. Although the SPoC eventually should cover the full lifecycle of businesses, we concentrate in this paper on the very beginning of the process, i.e., the opening of a new business.

In the following we do not claim to present an exhaustive list of requirements, but we concentrate on the most relevant ones: the key requirements are to (i) minimize the burden for citizens, (ii) guarantee that a certain time limit for the overall process of opening of a new business is not exceeded, and (iii) ensure the continuous monitoring of the process execution for the applicant.

To lower the burden for the citizen all required data should only be requested once. The individual application process instance has to be created, executed and monitored based on the collected data, in particular the data about the situation and the desired business registration, which essentially means that the SPoC has to know which kinds of permits have to be requested for a particular case and which concrete public authority or other institution is responsible for handling the request and granting the permit.

To minimize the duration for each application the process has to be optimized during the planning, i.e. process steps have to be parallelized as much as possible. This means that while one permit is being processed, another one which is independent of the first one can be processed at the same time. Essentially this means that the SPoC has to have knowledge about the dependencies of all process steps.

Several thousand public authorities are involved in handling licensing requests, and their involvement differs depending on the particular license – e.g., the processes for opening a café or a barber's shop differ substantially. Further, all these authorities have the legal power to design how they offer particular services. As argued above, the situations in which a citizen may be can vary substantially – e.g., for opening a barber's shop, suitable education is required; but which of the many educational offerings in today's 27 member states are perceived as suitable is a non-trivial question.

Thus, we assume that dealing with all these variations by manually designing all process variants or by means of configurable process models [10][11] is infeasible. Instead, a dynamic composition of the process is required where the individual situation of the applicant is mapped to the services from the various services providers, i.e. for Germany the local authorities. As we will see later, our approach makes use of knowing the input and output dependencies of each individual process step to compute an optimal composition into the overall process including a suitable parallelization of process steps.

As the SPoC also has to handle many requests at the same time, the composition of the individual process instances has to be automated. This allows the SPoC to concentrate on monitoring process execution and handle exceptions. This is important as a particular process instance has to be executed within a given time frame. Should the processing time exceed this threshold, then the request must be approved automatically. To avoid such cases, it is of necessity to equip the SPoC with monitoring options for the process execution.

Another challenging point is the question how the SPoC can be enabled to collect and maintain the knowledge about all authorities involved and their service offerings. Due to the high number of authorities and services, the most pragmatic solution is to collect and maintain the information in a decentralized way. A detailed discussion of this point is subject to our ongoing work, but out of scope for this paper.

In [1] a framework architecture for the realization of a single-point-of-contact in Germany is described. The architecture is very high level, as they want to allow for different specializations of this architecture e.g. in the different German federal states. In contrast, this paper describes one particular specialization of this framework architecture. The main difference lies in the way how processes that are executed by the SPoC are generated. In the related work they are pre-defined and can be adapted by ad-hoc workflows, if necessary for a specific case. In contrast to that we dynamically compose the processes making use of the Central Services and Administration Registries.

In general, our approach relies on the use of Semantic Web Services. A good overview of recent research is given in [8]. More specifically we compose services automatically based on goals, pre- and post-conditions such as described by the framework around the Web Service Modeling Ontology (WSMO) [9]. However, we herein pursue a systems approach, i.e., we make use of existing components where possible, and describe how the missing pieces can be designed. For instance, we use the composition approach described in [3], but do not make technical contributions to the composition itself. For a treatment of related work on other composition approaches, the interested reader is thus referred to [3].

Since central coordination is of high importance, we pursue an orchestration approach and do not consider autonomous agent technology [13].

1.3 Solution Approach

The desired SPoC collects the relevant data from the applicant only once for all process steps where it is required via a Web form which adapts on-the-fly according to entries of the applicant; on this basis the SPoC composes the optimal licensing process and computes its costs and duration; then the SPoC executes the process without intermediate delays, given all process steps perform as expected and all licenses are granted. Since all data is handled and forwarded by information systems, media disruptions can be avoided and the process execution time may be lowered considerably. In general, through the centralized view on the process, transparency in terms of operational relations and effort in terms of time and cost is increased.

In the next section we describe the main contribution of this paper: our overall approach for an implementation of the EU Services Directive. Section 3 discusses the proposed approach and concludes the paper.

2 Implementing the EU Services Directive

In this section we describe our solution approach along the lines of the behavioral view depicted in Fig. 1.

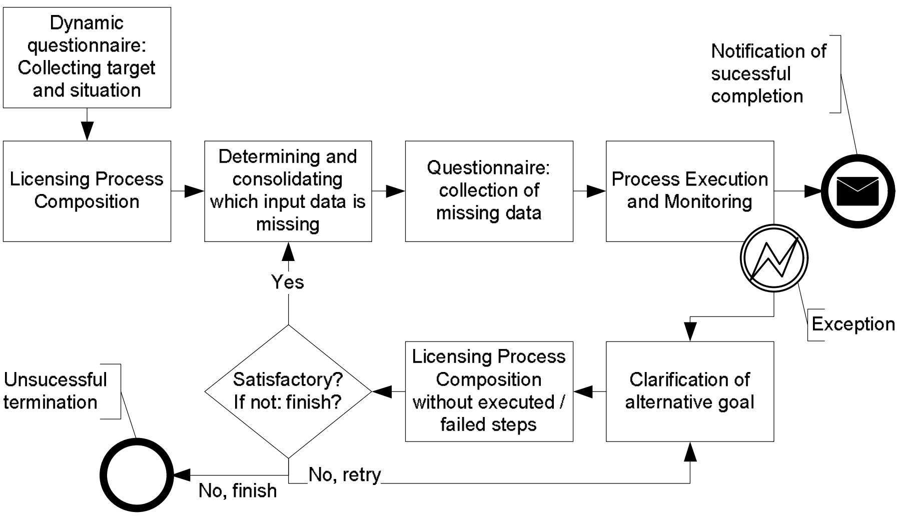

Fig. 1. Behavioral view of our solution approach

We give details on how the situation of the citizen is captured by dynamic questionnaires (Section 2.1), which enables the automatic composition (Section 2.2). The composed process then is displayed to the user (Section 2.3). From the process we can infer which information is required and not yet available, and the missing data can be requested from the user (Section 2.4). Once the data entry is completed, the process can be executed and monitored during execution by the stakeholders (Section 2.5). How to deal with exceptions during the execution is described in Section 2.6.

The descriptions of the behavioral view make use of components which are shown in the structural view of the proposed system, Fig. 2.

A user, be it a human agent of the SPoC or the applicant himself, interacts with the Web server of the SPoC through a Web browser for the purpose of requesting the license of his choice. This Web server provides two kinds of questionnaires: (i) the Dynamic Goal Questionnaire, which is used to enter the license type, its conditions, and the situation of the requestor; and (ii) Data Questionnaire for entering the missing data for the licensing steps. The former part is used for the data for the semantic goal, and the latter (potentially together with parts of the former) is the actual input data for the individual services. In the following, details of the individual steps are provided.

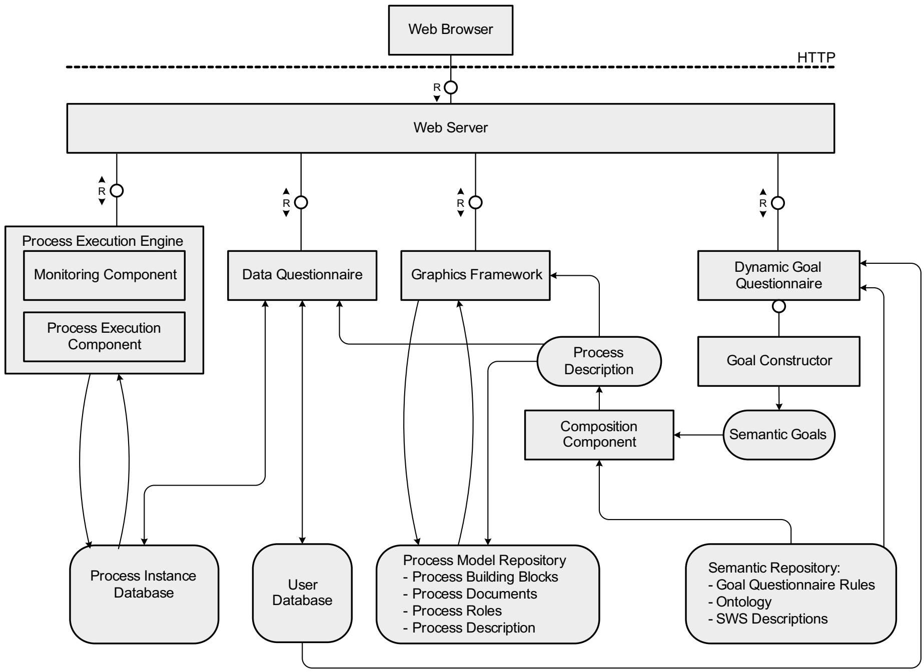

Fig. 2. Structural view of the suggested system (in FMC Component-Block notation[1])

2.1 Dynamic Questionnaire: Collecting Target and Situation

As a first step, the *Dynamic Goal Questionnaire* is filled in. Based on the *Rules for Questionnaire Creation*, the information that is relevant to the goal is requested from the user. After selecting the main license type (e.g., opening a Café in Cologne), these rules guide which information is relevant to the main license type, (e.g.: is the requestor interested in hosting life music performances or offering a food menu?). As this is an iterative procedure of data entering and rule evaluation we call it “dynamic”.

Once the goal is specified, the situation of the applicant needs to be entered (e.g., a Spanish citizen with education certificates from the Netherlands and an Italian wife). Parts of this information may already be available in the system, namely in the *User Database*. The question which information about the situation of the citizen is relevant for the licensing depends on the desired license. In fact, the data that may legally be collected by authorities may be restricted, again with respect to the desired license.

[1] http://www.fmc-modeling.org/

2.2 Licensing Process Composition

The input gained from the *Dynamic Goal Questionnaire* is taken as input to the *Goal Constructor*, which creates the semantic goal (i.e., according to WSMO) for this particular requestor. This input contains the information on the current situation of the requestor and the desired license including the license's relevant parameters. Abstractly speaking, the task of the automatic composition is to find a sequence of (atomic) services which, starting from the current situation of the requestor gets him to the target state of having the given license. The services here correspond to the licensing steps. The input information from the questionnaire is interpreted by the Goal Constructor, which constructs logical state representations of the current and the targeted state. In composition parlance, these states constitute the precondition (the current state as the situation of the requestor) and the postcondition (the target state) of the goal.[2]

With this goal, the *Composition Component* is invoked. As said, the task of the Composition Component is to find a sequence (or, more generally, an orchestration) of services which together can fulfill the goal. As the goal, the services are described in terms of their preconditions and postconditions (also referred to as effects). Note that a composition based on inputs / outputs only does in general *not* suffice to solve the problem at hand. This is due to the fact that the semantics of a service cannot be captured in inputs / outputs only[3]. Any Composition Component which can compose atomic services based on their preconditions and effects can in principle be used here. Examples for works on automatic composition of semantic Web services include [3][4][5], and are often based on AI Planning results, such as [6].

In particular, our prototype uses the implementation presented in [3]. In short, this composition approach builds on the FF algorithm [6] from AI planning, and extends it with specifics for SWS composition. It performs a forward search in the space of possible compositions. This search is guided by a suitable heuristic function, i.e., computing solutions to a relaxed composition problem which is exponentially easier than the full problem. Further, an ontology (with limited expressivity) can be taken into account. A purely sequential solution is composed at first, and parallelized as far as possible in a post-processing step. Due to the error handling procedure discussed in Section 2.6, there is no need for composing a contingency plan containing case distinctions for possible deviations from the desired outcomes of the process steps. The composition approach in [3] is particularly well suited for the scenario presented in this paper, since it scales up well to large service repositories. However, other approaches could in principle be used too.

Regardless of the specific composition approach used, the Composition Component takes the available *Semantic Web Service (SWS) Descriptions* and the *Ontology*[4] from the *Semantic Repository*, and automatically composes a *Process Description* for the requestor. The *SWS Descriptions* are abstract descriptions of the Process Building

---

2 Note that the precondition is perceived as part of the composition goal.

3 The unconvinced reader may consider a service with a car description as input and a price as output. Does the service buy cars, sell cars, or assess the value of cars?

4 The ontology serves as a model for vocabulary and relations in the domain of discourse. The approach can also make do if the relations are not modeled (or empty), or if the vocabulary is presented in some other form. For our solution we have engineered an appropriate ontology together with domain experts from the SAP Public Sector field organization.

Blocks which reside in the *Process Repository*. In SWS parlance, the *SWS Description* contains a "grounding" to a *Process Building Block*. Note that the *Process Building Blocks* are not required to be implemented as Web services, but we describe them using SWS standards. Note further, that the differentiation between the two repositories in the architecture is a logical one, i.e., they may be implemented as a single system; this also impacts the pragmatics of life-cycle management of the *Process Building Blocks*, which again is outside of the scope of this paper.

There is a notable difference to standard workflow approaches, where a model is created once and executed in many instances; here, the model is created *for* the instance and subsequently executed.

The *Process Building Blocks* may specify the following additional information (i.e., in addition to their precondition and effect):

- The required input data, e.g., name, maiden name, address, date of birth, passport number,
- the actors who are involved in performing the activity, e.g., the central federal registry of Germany (de: "Bundeszentralregister") is responsible for creating police record excerpts,
- the maximum duration, e.g., two days, and
- the price, e.g., 10€.

In order to give overview information on the process level to the user, the respective sums of the latter two aspects are computed. Since a shorter duration of the licensing process is desired, the composed process is (ideally) parallelized to a maximum degree – i.e.: if a step in the process is not dependent on some other step (directly or indirectly), or vice versa, our system assumes that those two steps can be scheduled for parallel execution. Depending on the exact execution semantics of the process execution language, the maximal process duration may be calculated as the sum of durations of the longest sequence of contiguous steps in the process. The sum of the cost can be computed in an easier way: it is simply the sum of all costs of *Process Building Blocks* in the process.

A screenshot of the composed process is shown in Fig. 3 where each step of the process is shown as a blue box, starting from the top going to the bottom where parallel processes do occur (to minimize duration of the overall process). For the step *Assign Tax Number* a yellow box, which appears with a mouse over, shows the annotated meta data including the actor, the pre- and post-condition (internally represented according to WSMO), the time and the costs.

2.3 Presentation of the Composed Process to the User

The composed process then is persisted in the *Process Model Repository* for later use, e.g., through the *Process Execution Environment*[5]. Next, the *Process Description* is displayed to the user on a Web site, e.g., in the form depicted in Fig. 3, by using a *Graphics Framework*.

---

[5] A standard WSBPEL [12]execution engine or a workflow execution engine can be used as execution environment. If the format in which the process model is represented is not interpretable by the execution engine at hand, a translation needs to take place. In this paper we neglect the details of the format, since the specifics have no impact on our approach.

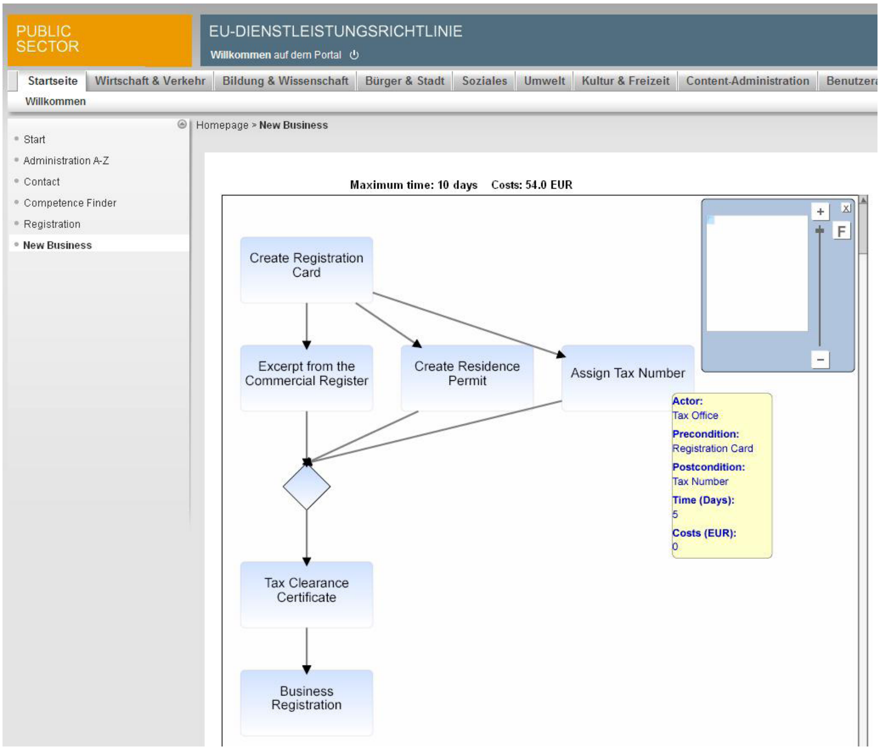

Fig. 3. Screenshot of the prototypical implementation: automatically composed process for opening of a Café of a Spanish citizen in Germany

Note that this view may only be appropriate if the user is a representative of the SPoC; if the user is the applicant, then this view may be hidden. Otherwise, the *Graphics Framework* automatically creates a layout for the composed process model and renders the graphical representation into a format that can be transmitted by the *Web server* and displayed in the *Web browser* of the user. Our prototype, e.g., uses support vector graphics (SVG) for this purpose.

2.4 Consolidation of Input Data and Collection of Missing Data

Once the *Process Description* has been rendered and graphically displayed to the user, he may select to start the execution of the process, in which case it first has to be clarified if all required input data is available. This is done by iterating over the composed process steps and collecting the required input data for all of them. This collection then is consolidated, i.e., it is checked if the same data is used by multiple steps (e.g., name, address of the requested pub, home address). The consolidation can be based on data standards, e.g. UN/CEFACT CCTS [7]. Subsequently, it can be compared (using the same technique) whether the consolidated input data is already available in the system (in the *User Database*), or whether it still has to be entered.

If there is missing data, it is requested through the *Data Questionnaire* from the user. The data gained here can be stored in the *User Database* - if legally permissible and agreed to by the user. As for the licensing process, the goal is to collect all necessary data upfront, once, and to progress the licensing process afterwards without intermediate delays. This feature by itself is already quite a significant benefit for the requestor, as it speeds up the process and massively lowers the burden of filling in forms in comparison to today.[6]

2.5 Process Execution and Monitoring

Finally, when the process model and all input data are available, the process execution can be started. The execution is mostly steered by the *Process Execution Environment* (cf. footnote 5). The progress made can be reviewed through a *Monitoring Component*, while the actual process control remains with the *Process Execution Component*. During the execution, *Process Instance Data* is persisted in the *Instance Database*, also for purposes of potential auditing later on.

2.6 Exception Handling

Since many of the process steps are requests for granting certain permissions or the like, there is a chance that not all of them complete successfully – e.g., the permission for live music performances may not be granted for locations in quiet residential areas. For such exceptions, the system allows to *re-plan* the process, given the new information. The requestor is pro-actively notified (e.g., via e-mail) of the changed situation and may choose an alternative goal; the options are presented through the *Dynamic Goal Questionnaire* again.

If an alternative goal is chosen, the *Composition Component* is invoked with a new goal from the *Goal Constructor* together with the current state of the process. If a satisfactory solution can be found this way, the procedure is restarted at the point where the input data is collected. Additional data that was not required before is collected from the requestor at this stage. In contrast, if no satisfactory solution can be found for the user, even with multiple attempts, the process is terminated in an unsuccessful state.

In contrast, if all licenses are granted or occurring exceptions are handled satisfactorily, the applicant is informed about the successful outcome of his request. He now may start doing a hopefully successful business in Germany.

3 Conclusions and Outlook

In this paper we presented a conceptual architecture and a running prototype for implementing the key requirements imposed by "EU Services Directive" to the European member states. In a nutshell, these requirements are: to offer a Single-Point-of-Contact for distributed public services; the possibilities to intelligently create license processes for citizens; to pro-actively inform citizens about required changes in the process due to newly available information; and to allow citizens to remotely monitor the status of their inquiries.

[6] Anecdotic evidence tells us that more than one hundred forms may have to be filled in today in order to open a barber's shop in Germany today (given the requestor is a non-German EU citizen who does not live in Germany yet). Unfortunately we could not verify the anecdote so far due to time constraints.

At the core of the solution is a composition component which can, based on a semantic goal, create a licensing process tailored to the situation and objectives of an individual requestor. In order to enable the use of this approach, a dynamic questionnaire collects the relevant information on the current situation of the requestor as well as of the specific conditions of the requested license. Subsequently the automatic composition of previously specified process steps, or services, can be performed. Once the process has been composed, the system checks if the input data for all licensing steps in the process is available, so that the process can be executed without intermediate delays. If that is not the case, the missing input data is collected. When all data is available, the execution is started.

Due to the nature of the process steps it may well be that some of them deliver a negative outcome, e.g., if a license is not granted. In this case, our approach helps in dynamically re-planning for the changed situation, e.g., suggesting alternatives to some desired license conditions or suggesting a different (potentially more pricey) way to achieve the original goal.

Based on the upcoming necessity to automatically grant licenses as soon as the processing time exceeds the threshold, automatic monitoring seems to be a desirable feature for the SPoC. In the future, this may be implemented on the basis of the solution presented here. Further, the SPoC provides most benefit to all EU citizens if the data entries may be made available in multiple languages – be it through human agents working for the SPoC, or through forms on Web sites, where the effort of translation has to be spent only once. However, this also has been out-of-scope for the initial prototype but will be considered in later versions.

The EU member states who are leading in providing their citizens electronic access to public services usually “only” offer access to electronic forms which can be printed and filled in at home, along with descriptions of public processes to guide citizens. The next wave will be the provisioning of all kinds of public services as electronic services, including all services that are needed to implement the EU Services Directive.

While the presented approach is designed for the particular setting of the "EU Services Directive", we see it as the blueprint for various scenarios of composite citizen-centric services. First discussions with colleagues from the Public Services field organisation at SAP indicate that our solution is broad enough to support many different kinds of public services – but this still has to be proven. Key questions for future research are: how to ensure completeness of the created processes; how to achieve required coverage of public sector processes; evaluate the feasibility of semantic service annotation and consider automated approaches for service annotation; consider security aspects for public services.

References

[1] Further information on the EU Services Directive (observed on 2009-02-02), `http://ec.europa.eu/internal_market/services/services-dir/index_en.htm`

[2] Lucke, J. v., Eckert, K.P., Breitenstrom, C.: EU-DLR - Einheitlicher Ansprechpartner: Rahmenarchitektur und technischer Lösungsvorschlag. Whitepaper, Fraunhofer-Institut für Offene Kommunikationssysteme (FOKUS) (February 2008)

[3] Hoffmann, J., Weber, I., Scicluna, J., Kaczmarek, T., Ankolekar, A.: Combining Scalability and Expressivity in the Automatic Composition of Semantic Web Services. In: ICWE 2008: Intl. Conf. on Web Engineering (July 2008)
[4] Meyer, H., Weske, M.: Automated service composition using heuristic search. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 81–96. Springer, Heidelberg (2006)
[5] Constantinescu, I., Faltings, B., Binder, W.: Large Scale, Type-compatible Service Composition. In: International Conference on Web Services (2004)
[6] Hoffmann, J., Nebel, B.: The FF Planning System: Fast Plan Generation Through Heuristic Search. Journal of Artificial Intelligence Research (2001)
[7] United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT). Core Components Technical Specification. Part 8 of the ebXML Framework (November 2003), http://www.unece.org/cefact/ebxml/CCTS_V2-01_Final.pdf
[8] Studer, R., Grimm, S., Abecker, A. (eds.): Semantic Web Services - Concepts, Technologies, and Applications. Springer, Heidelberg (2007)
[9] Bruijn, J.d., Fensel, D., Keller, U., Lara, R.: Using the Web Service Modeling Ontology to Enable Semantic E-Business. Communications of the ACM 48(12), 43–47 (2005)
[10] Gottschalk, F., Van Der Aalst, W.M.P., Jansen-Vullers, M.H., La Rosa, M.: Configurable Workflow Models. International Journal of Cooperative Information Systems (IJCIS) (2008)
[11] Rosemann, M., Van Der Aalst, W.M.P.: A Configurable Reference Modeling Language, Information Systems. Elsevier, Amsterdam (2007)
[12] Alves, A., et al.: Web Services Business Process Execution Language version 2.0, Committee specification, OASIS (January 2007)
[13] Russel, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Prentice-Hall, Englewood Cliffs (2002)