ACCESS-EGOV – A REAL-WORLD SEMANTIC SERVICE-ORIENTED ARCHITECTURE FOR E-GOVERNMENT

Rolf Schillinger, Stefan Dürbeck[1], Peter Bednar[2]

Abstract
Service-oriented architectures have climbed up the path from theory to being widely implemented for use in productive computing environments in both eCommerce and eGovernment. In the latter field of application, new tasks and challenges have arisen from the European Union's Services Directive, entailing on-demand collaboration of back-office services. Despite the striking success that cooperating web service landscapes have experienced over the last couple of years, there are still serious shortcomings that hinder further propagation of SOAs for collaborative ad-hoc computing scenarios, such as to sufficiently deliver high-quality services in the eGovernment sector. Based on our practical experience in the EU-funded research project Access-eGov, we outline the shortcomings of traditional SOAs, prove the necessity of introducing semantic technologies and show how ontologies and semantically enriched process workflows can be utilised in order to mitigate them.

1. Introduction

Public administrations, especially in European Union member states, today face far more technical challenges arising from legal and organisational regulations, than it was the case during the beginning of the electronic data processing age some decades ago [10]. A lot of these newly arising challenges have their origin in legal frameworks, such as pan-European EU directives; other ones are emerging from a lately applied economic view, considering public administrations as cost-driven service providers that also need to save on expenses, e.g. on the technologies that are implemented on-site.

The European Union's Services Directive as of December 2006 is envisioning a new level of cross-boundary service delivery in the private and public sector alike and is therefore one particular important challenge for public administrations. It intends to help developing a pan-European single market in the services sector by tearing down barriers to cross-border trade between the member states, thus e.g. facilitating to allow businesses to provide well-defined services in other EU Member States.

[1] University of Regensburg, 93053 Regensburg, Germany, Universitätsstraße 31
2 Technická univerzita v Košiciach, 042 00 Košice, Slovakia, Letná 9

In addition to strengthening the pan-European eCommerce sector, the Directive implies the concept of on-demand collaboration of public agency services, triggered by citizens or businesses, and all moderated under a single leading agency (called Point of Single Contact, PSC) that is hiding the process complexity from the service requester as for example noted in [7]. Every member state government of the European Union is required to have the hereby mentioned organisational means implemented by December 2009 that allow citizens to interact with public administrations in a whole-of-government manner.

In detail, [8] describes, that the Services Directive requires Member States to:

- offer PSCs through which citizens and businesses will be able to find the information and complete the administrative formalities necessary for doing business in their country or legal domain,
- create citizen confidence in cross-border service delivery through access to information and high quality of service,
- abolish restrictive laws and administrative practices that prevent commercial service providers from offering services across national borders and
- allow for easier cooperation between authorisation and regulatory bodies across the European Union, thereby reducing administrative barriers for citizens and businesses.

Many governmental information system bodies consider the broadened use of service-oriented architectures as possible approach to hide the complexity of back-office interaction caused by meeting the requirements of the EU Services Directive [11]. Especially the case of cross-organisational service delivery raises problems arising from the semantic differences with which service types and categories are annotated in different legal domains.
Access-eGov, a research project funded under EU's FP6 programme, aims at supporting public administrations in such a cross-organisational service delivery, as well as providing citizens and businesses with an elegant and usable web agent portal through which they can trigger public service activities on their behalf. At the core of the research work that has been undertaken is a service-oriented architecture which is enhanced by semantic technologies in order to provide semantic mapping of service descriptions across organisational and legal boundaries and execution of pan-European service delivery scenarios, as envisioned by the EU's Services Directive as described above.

In the following chapters, we describe the traditional basics of service-oriented architectures (section 2) and also lay down the key concepts that a semantically-enriched SOA shall live up to in order to mitigate these shortcomings of past SOA implementations. The novel integration of a semantic-enabled process model for orchestrating public agency (web) services within the scope of Access-eGov, is later on presented on a detailed level in section 3. The progress that has been made towards building a semantic service-oriented architecture for Access-eGov is outlined in section 4. Finally, work conducted by researchers and practitioners from related EU-funded projects is also outlined in the respective chapter (section 5).

2. Semantically enriching Service Oriented Architectures

Access-eGov is all about adding semantic information to traditional SOAs, yielding as result a Semantic Service-Oriented Architecture (SSOA). The considerations in this chapter have led us to believe, that SSOA is the future in service orientation while traditional SOAs do not offer a suitable added value to justify their high complexity and costs.

2.1. SOAs compared to traditional components

In [6], Erl summarises key concepts in a concise manner and finally offers a list containing loose coupling, service contracts, autonomy, abstraction, reusability, statelessness, composability and discoverability. If looked at in detail, most of these key concepts are interdependent on others: If services are loosely coupled, meaning they can easily be connected without many interdependencies, this is done through the use of service contracts (i.e. interfaces) which abstract the services' inner workings. Loosely coupled entities can be autonomous (useful on their own) which also makes them reusable. The lack of any state information in calls to such entities only furthers the degree of loose coupling between them.

The implementation of these key aspects is not restricted to the SOA world. Almost all of the above mentioned concepts can be implemented using traditional component models (e.g. CORBA, COM+) or even, on a lower level, by a careful design of objects or modules. Two exceptions exist which are explained in the following.

Composability, meaning the possibility to order existing services in new ways with the aim of offering new, dynamically created functionality cannot easily be implemented by traditional methods which are not geared towards reusability on a service level. Even those composition techniques, that currently are available in SOAs cannot be supported by tools; it is a highly intellectual task with lots of involvement of developers [5][9][16].

Discoverability as the basis for composability has no direct match in traditional components either. The usual way of "discovering" services again is through developer intervention by searching for fitting functions, modules or objects (according to the chosen development model) in catalogues, registries, specialised portals or search engines as shown in [3]. A worst case scenario, appearing quite frequently, leaves developers searching for reusable parts in the source code if none of the above mentioned means are available.

Traditional SOAs offer only limited cure for composability and discoverability issues. The composition process in SOAs is as complex as in traditional software engineering settings. Standardization of service descriptions in WSDL is the only support a SOA has to offer. While this standardization is very important, using only information contained in WSDL descriptions disallows any tool to do a very simple composition process on its own because there is no automated way of understanding the meaning of the service description and therefore no way of understanding the effects of executing that service.

The underlying discovery process suffers from the same problem. Automatically searching for services with given effects/outputs is impossible since no tool is able to understand the functionality of the service queried which in turn is caused by service providers having no opportunity to describe their services in a machine readable or at least structured form.

2.2. Semantics and SOAs

Since not all of the above mentioned shortcomings are easily solvable, adding - possibly complex - semantic information to the web service descriptions paves the way for working discovery and composition.

In general, semantic information is stored in some sort of knowledge storage (usually an ontology), a formalism for transforming the knowledge storage into something transportable (most often at

least a possibility to transform the data into XML exists) and an accompanying toolchain with – among others - tools for importing and exporting knowledge, for semantically enriching existing web service descriptions or for generating new semantic service descriptions. The toolchain is highly specific to the current task and therefore almost never part of the semantic framework. A number of such frameworks currently exist, however careful examination in the initial project phase showed that only three of them were suitable for Access-eGov as they were the only ones being able to work with web services without extending their core concepts.

The Web Service Description Language - Semantics (WSDL-S) [1] has no built-in way of describing semantic annotations but rather provides means of linking semantic annotations to standard WSDL web service descriptions. Access-eGov did not choose WSDL-S because orchestration, choreography, preconditions and effects (which we consider most important) are not directly supported.

OWL-S, an ontology based on the Web Ontology Language (OWL) [15] is a different concept. Top level entities described by OWL-S are a ServiceProfile (the service's functionality), a ServiceModel (information on the composition of services) and a ServiceGrounding (information about invocation of services). While the OWL-S process model (it uses atomic and composite services, just as needed in Access-eGov) has a certain appeal, Access-eGov opted against using OWL-S because the proposed method of using OWL as a glue between different rule-based languages such as the Knowledge Interchange Format (KIF) could lead to undecidability of the resulting expressions.

The Web Service Modelling Ontology (WSMO) [2] uses the core concepts Ontologies, Web Services, Goals and Mediators to describe the world of discourse. WSMO has a very mature design despite being comparatively young. It was created from scratch without legacy dependencies. Many features lead the Access-eGov group to evaluate WSMO as the best fitting framework. What contributed to this decision are the strong development group behind it, the layered logical languages that constitute the WSML family, the mediation concept and already existing mappings to other languages.

2.3. Semantics in the light of the EU service directive

Traditional SOAs are of limited use for eGovernment when it comes to executing web services in cross-border scenarios (as intended by the Services Directive). Here, service descriptions are only sometimes published using domain ontologies that are often of proprietary nature as well. These circumstances require the use of semantic metadata annotation that can be used for automated mapping of ontological concepts between different organisational domains.
In order to promote system interaction based on semantic metadata, a number of efforts have been undertaken at national levels throughout the European Union member states, to ensure data can be uniformly processed by as many parties as possible that are tied together in public sector process chains, regardless of the organisational boundaries. Currently, there is no union-wide public sector ontology available, but in the run-up to it, a number of government vocabularies are being elaborated that are supposed to form the basis of a metadata ontology. Access-eGov is therefore working with the LEIKA, the German standardised services catalogue that has been constructed by the federal government initiative "Deutschland Online" bearing the Services Directive in mind. Featuring all relevant public services that can be offered through a PSC, it is still work-in-progress and thus not officially available to this date.

3. Access-eGov

3.1. Overall Access-eGov Architecture

Figure 1 describes the core system components of the Access-eGov platform. Interfaces to the system are depicted as being XML-based web services. Every user of the platform, including the actual GUI, the so-called “Personal Assistant Client” (see section 4.1) connects to this endpoint.

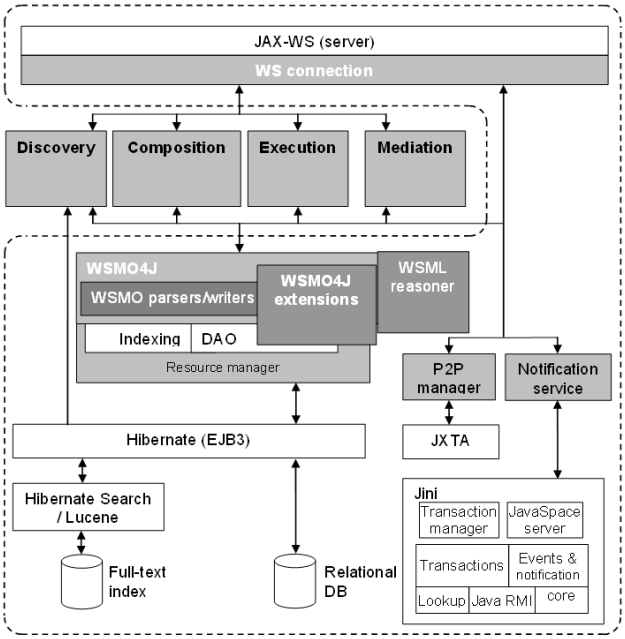

Figure 1: The Access-eGov core architecture

A standard object-relational mapper (Hibernate plus caching facilities) is used to store all of the entities involved in a relational database, acting as a repository. Hibernate also allows for easy registration of trigger-like event handling (OnUpdate, OnInsert, …) which we use to connect the database backend to a search engine. This full-text search was no architectural choice, but a user requirement.

A specially-tailored implementation of WSMO for Java (WSMO4J) is at the very heart of the platform, contributing the implementation of the whole object structure of WSMO. First trials of our repositories with the original wsmo4j distribution did yield performance problems when used in conjunction with the hibernate setup. It turned out, that the inheritance hierarchies involved in the wsmo4j code were the show stoppers. Currently there are two types of parsers or serializers for WSMO, namely a WSML one and a XML one. Due to reasons explained in section 3.2 we had to add certain extensions to WSMO which are also implemented in the extended version of WSMO4J we tailored.

The reasoner is another very important component as the discovery component relies on it for semantic matching. In the Access-eGov case, semantic matching means that certain service properties are compared to requester requirements raised by the specific goals of a requester. The properties in question are called "capabilities", a semantic construct that has been introduced by WSMO, providing information on the functionality offered [2]. The introduction of the reasoner component encapsulates this strategy, therefore making it interchangeable. Currently different strategies are tested in the pilot projects.

Core functionalities in this platform are performed by the following components:

- Discovery to find out which services fit the postulated capabilities
- Composition to build an invokable chain of services from the users' goals
- Execution of web services
- Mediation to act as an intermediary between different ontologies.

Access-eGov's main usage lies in the brokering of services that potentially last very long (weeks, up to months), making it necessary for the platform to offer some kind of notification feature that notifies interested parties of certain events as well. For performance and safety reasons, on-site Access-eGov platform instances, so called AeG-nodes, will be distributed over a small-scale peer-to-peer network or possibly the internet itself.

3.2. Access-eGov process model

The process model in Access-eGov will be used to guide citizens to achieve specific goals as well as to coordinate activities performed by all actors - citizens, traditional public administration services and web services. The current WSMO specification for the process model is based on abstract state machines (ASMs), constituted by state signature represented by the ontology, and *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 the other.

We have found that the modeling of the processes using the ASMs is less intuitive and that the current proposal of the WSMO specification is not suitable for Access-eGov. It appears to be too complex and non-intuitive for the target audience of public officers who might be domain experts but no knowledge modeling experts. Since the specification of the process model is not finished, the dataflow is unspecified in certain situations. Trying to solve these particular shortcomings, we devised a new, workflow-based process model which is described in every detail in [18].

4. Real World Application

Based on the technological and theoretical fundaments as explained beforehand, Access-eGov rolled out its program suite to the pilot partners during the pilot phase in mid-2007 and again in autumn-2008 enhanced by major feature improvements. A detailed evaluation report of the first trial can be found in [20]. This suite also includes administration tools for domain experts in the respective public administrations in order to facilitate day-to-day handling of the semantic service descriptions and to guarantee a certain ease of use to configure the services platform.

4.1. Personal Assistant Client

A graphical user interface, called the "Personal Assistant Client" (PAC) is available through the pilot phase for public use in Schleswig-Holstein/Germany, Slovakia and Poland. It enables citizens to take part in the evaluation and to make use of the semantically enriched (web) services brokered through Access-eGov. The user interface dynamically generates a process workflow model, given certain user input regarding the current life situation, like "Applying for marriage" in the German or "Registering a company" in the Polish pilot trial. The offered web services are then retrieved using the domain ontology in a legal administrative domain (e.g. Schleswig-Holstein).

The PAC interface then uses Access-eGov core components to generate a list of user actions based on the semantic service annotations. This list of service steps are to be undertaken in order to trigger a complex service chain, also containing web services if offered by public administrations.

In our conception of Access-eGov, the Personal Assistant Client interface can act as Point of Single Contact for citizens wishing to trigger public services that are offered across organisations as (web) services. In the case of the field test installation in Schleswig-Holstein, the Personal Assistant Client installation hosted by our user partners at the Ministry of Finance acts in the very sense of the EU Services Directive, as PSC for subordinate administrations within the boundaries of this German State.

4.2. Annotation tool

In order to feed semantic descriptions of the offered web services into the Access-eGov service repository, an easy-to-use graphical user interface (a so-called "Annotation Tool") was obligatory to be shipped for public servants administering the offered services. Since (web) service descriptions may vary, according to the legal and organisational core concepts that are underlying the respective governmental system, the user interface for public administrations also differs from region to region. A typical service description that was found out to be differing from one legal domain to the other one was, whether or not a direct contact person at the public agency needs to be given who is held responsible for service delivery.
This discrepancy made it necessary to abandon static front-ends for public servants, but to set up a system that dynamically creates annotation forms from ontologies on-the-fly. During the trials conducted in the three user partner regions (Schleswig-Holstein/Germany, Slovakia and Poland), most public administration domain experts found the ontology-driven Annotation Tool helpful and easy-to-use after a short introductory training course.

4.3. Current status and future work within the project

For the course of the last 2.5 years, Access eGov has been able to build a Semantic SOA from both stock open-source components and own extensions. If looked at from the angle of objectives as set out by the consortium before the project onset, Access-eGov proves to be very successful. The following overview shows the objectives as outlined in the Technical Annex to the project contract:

Reference ontology
So far we have created ontologies for every pilot which is the best approach in a pilot setting [15]. Turning these into reference ontologies needs careful networking with other projects which will be the last step in the Access-eGov ontology development cycle.

Semantic mark-up
With the Annotation tool as described in section 4.2, we have provided the user partners with an intuitive and practicable way of annotating their services. Packaging such a difficult matter in an application proved to be very complex which can be seen by the fact, that development of the tool is still not finished. Initially it was envisioned, that semantic annotations also be possible through the users' standards tools like CMSs but it turned out to be way out of the scope of a research and development project as Access-eGov is. The exploitation phase commencing after project end however will rely on creating plug-ins to existing tools in order to be successful.

Tools for discovery
Access-eGov now constitutes a mature platform for semantic discovery. After struggling with performance issues caused by interactions between the inherent complexity of the original WSMO4J distribution and the storage facility, the refactoring of that library was successful and lead to a system with good performance. A potential for fine tuning the system exists, this was examined through extensive benchmarking.

Composition
The prototype already exhibits the features needed for real-world usage as it is able to performantly and correctly resolve a generic workflow to a specific, executable construct.

Distributed Security Infrastructure
Delivering a distributed security infrastructure proved to be a daunting task. [13] and [14] presents the results of the Access-eGov project's research into that area. It describes an architecture similar to the one specified in the well-known XACML specification by OASIS. Additionally the model introduces a Certificate Authority based schema to create trust between the main actors. This trust allows for easy distribution of central components of this infrastructure between different service providers, helping gain privacy, security and performance.
Extending this schema, a prototype is currently implemented that uses the Access-eGov infrastructure to semantically compile executable "security processes" from abstract security scenarios which are comparable to the scenarios described in section 3.2.

5. Related Work

The topic of enabling public administrations in the European Union to interact with each other on behalf of citizens requesting certain services in electronic ways, has been occupying researchers and practitioners for more than a decade now. A number of successful research projects stated the problems that they encountered when trying to electronically streamline the whole eGovernment process chain and thus tried to focus their work on specific areas on the long way from the citizen via the service provider through to back-office administrative processes. One of them, the EU-funded project eGOV concentrated on the user side and created an integrated portal platform for citizens to get information about public services [22]. Access-eGov took up this user-centric approach in the Personal Assistant Client, but takes the idea further to act as PSC, hiding process complexity in the Services Directive sense.

As opposed to many EU-funded eGovernment projects which produced public sector domain ontologies as a by-product tailored to the specific project settings, the OntoGov project, described in [19], focused on generating controlled vocabularies that are generic enough to be re-used in further projects and across organisational domains. Although these ontologies covered all aspects of managing the life-cycle of services within public administrations (like composition, re-configuration and maintenance), they proved to be of limited value to our specific setting only since OntoGov's model for incorporating traditional (offline) services differs entirely from the Access-eGov approach [18].

Our research in Access-eGov is based on the ontological concepts of WSMO and taking the work of this project one step further. WSMO, a semantic web service execution platform and an EU-funded research project as well, was initially not considering the idea to build service chains using generic workflow descriptions that will be tailored at execution time [21]. WSMO itself describes compositions using above mentioned abstract state machine transitions that proved to be far too complex to maintain in the highly dynamic scenario of ad-hoc (web) service customization.

Another project tied to semantic e-Government is Semantic Gov [17]. It focuses more on building a complete infrastructure for public administrations' semantic web service needs. Sharing the same start date, the projects could not benefit from each others' research in the very important design phase. SemanticGov's latest research results however highlight the fact, that indeed both projects took a diverging path with SemanticGov building the envisioned complex infrastructure with a strong focus on all semantic issues while Access-eGov took a much broader approach (e.g. using generic process models) as described above.

6. Conclusion and future work

We laid out the shortcomings that are inherent to traditional service-oriented architectural approaches, and showed ways how to mitigate them by introducing semantic annotations into web service landscapes. Based on the practical research work undertaken by the EU-funded project Access-eGov, we faced additional requirements in the public sector that were arising from current legal requirements, also having technical influence on SOAs in ad-hoc web service chaining scenarios. To our understanding, such requirements to cross-organisational service chaining (as of the EU Services Directive) are better to be dealt with by introducing semantic technologies into web service brokering, as is the case in Access-eGov.

The experience the consortium was able to build up so far will be the basis for some more publications. The consortium for example plans, to publish their findings on large scale benchmarks of semantic infrastructures.

The valuable research work from Access-eGov will be serving as experience basis for adjacent research project work in EU's FP7 research project SPIKE, aiming to take the idea of ad-hoc collaboration between web services one step further, by allowing whole organizations to temporarily cooperate over a longer period of time by securely linking their already existing web service landscapes.

7. Acknowledgment

The work reported in this paper was in part funded by the European Union within the project Access-eGov (Sixth Framework Program, Contract No. FP6-2004-27020). We would like to thank our project partners for all helpful comments and stimulating discussions.

8. Reference List

[1] Akkiraju R. et al, "WSDL-S Web Services Semantics---WSDL-S", W3C Member Submission , 2005

[2] Arroyo S., Bussler C., Kopecky J., Lara R., Polleres A., Zaremba M., "Web Service Capabilities and Constraints in WSMO", W3C Workshop on Constraints and Capabilities for Web Services, 2004

[3] Bachlechner D., Siorpaes K., Fensel D., Toma I., "Web Service Discovery-A Reality Check", Proceedings of the First Workshop: SemWiki2006-From Wiki to Semantics, co-located with the Third Annual European Semantic Web Conference (ESWC'06), 2006.

[4] Berners-Lee T. and Lassila O. a, "Web Semantic", Scientific American – American Edition, Vol. 284, part 5, pages 28-37, 2001.

[5] Curbera F., Ferguson D., Nally M., Stockton M.L., "Toward a programming model for service-oriented computing", Proceedings of 3rd International Conference on Service Oriented Computing (ICSOC'05), 2005

[6] Erl T., "Service-oriented architecture", Prentice Hall, 2004

[7] European Commission, "DIRECTIVE 2006/123/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 12 December 2006 on services in the internal market", Official Journal of the European Union, 2006

[8] H.M. Department for Business, Enterprise & Regulatory Reform, “Implementing the Services Directive, Government response to Consultation on implementing the EU Services Directive in the UK”, 2008
[9] Hanson J., “Coarse-grained interfaces enable service composition in SOA”, JavaOne Conference, 2003
[10] Janssen M., Gortmaker J. , Wagenaar R. W., “Web Service Orchestration in Public Administration: Challenges, Roles, and Growth Stages” Information Systems Management, 2006
[11] Klischewski R., Ukena S., and Wozniak D., “User Requirements Analysis & Development/Test Recommendation”, Access-eGov deliverable D2.2, 2006.
[12] Klischewski R., Ukena S., Furdik K., “Public administration resource ontologies “, Access-eGov deliverable D7.1, 2007
[13] Kolter, J., Schillinger, R., Pernul, G., “A Privacy-enhanced Attribute-based Access Control System”, Proc. of the 21st Annual IFIP WG 11.3 Working Conference on Data and Application Security (DBSec 2007), 2007
[14] Kolter, J., Schillinger, R., Pernul, G., “Building a Distributed Semantic-aware Security Architecture”, Proc. of the 22nd IFIP TC-11 International Information Security Conference (SEC 2007), 2007
[15] Lausen H., Polleres A., Roman D., “Web Service Modeling Ontology (WSMO)”, W3C Member Submission, 2005
[16] Papazoglou M.P., Georgakopoulos D., “Service-Oriented Computing”, Communications of the ACM, 2003
[17] SemanticGov consortium, “WP02 – Requirements analysis for national and pan-European e-Government services. D2.2 – Legislative, Institutional and EU Policy Related Requirements”, 2006
[18] Skokan M., Bednár P., “Orchestration of Public Administration Services with a Use of Semantic Technologies”, The 6th Slovakian - International Symposium on Applied Machine Intelligence, 2008.
[19] Tambouris E., Gorilas S. et al., “Ontology-Enabled E-gov Service Configuration: An Overview of the OntoGov Project”, Lecture Notes in Computer Science - Knowledge Management in Electronic Government, 2004
[20] Ukena S., Hajdukova G., Kleimann M., Sroga M., Bednar P., Hreno J., Dürbeck S., Gizikis A. “Evaluation of trial and specification for revision of components“, Access-eGov deliverable D8.3, 2008
[21] WSMO Project, “D2v1.2. web service modeling ontology (wsmo)”, Technical report, 2005
[22] Wimmer M. A., “A European perspective towards online one-stop government: the eGOV project”, Electronic Commerce Research and Applications, Volume 1, Issue 1, 2002