Test Strategies for Evaluation of Semantic eGovernment Applications

Ralf Klischewski and Stefan Ukena

German University in Cairo, Egypt
ralf.klischewski@guc.edu.eg, stefan.ukena@gmx.de

Abstract. In this paper we present a framework for identifying the test focus and test objectives based on the assumption that automatic information processing based on encoded meaning is the core of semantic e-government applications to be evaluated. Taking into account test strategies from software engineering and IT project management as well as different stakeholder perspectives, we identify possible test instruments. Several of these instruments have been applied in the Access-eGov project and we discuss the experience gathered in view of the newly developed framework in order to identify lessons learnt as well as to point to future research. The contribution of the paper is a portfolio of test strategies suggesting certain instruments to be applied from a systems view, agent view, and user view. We conclude that improving semantic e-government applications could be supported through applying a test-first approach, e.g. through providing an e-government test agent to be used in test labs or within the development process.

Keywords: semantic e-government application, test instruments, semantic interoperability, agent-based testing.

1 Introduction: "You Can Only Improve What You Can Test…"

The vision of the "Semantic Web" has received considerable attention also in the area of e-government: semantic interoperability is on the agenda of interoperability frameworks, and the use of semantic technology (which "encodes meanings separately from data and content files, and separately from application code"; Wikipedia) is expected to enhance the integration and quality of e-government services. In this context, we consider e-government applications as *semantic* applications if the computer-based provision of administrative information and services depends on explicit modeling of semantic concepts and the strategic use of semantic technology.

Meanwhile, a number of (research) projects have induced prototypes and pilot applications that can be classified as semantic applications in the above sense (cf. e.g. Abecker et al. 2006). However, applying as well as testing semantic technology and solutions in the area of public administration remains quite a challenge (cf. Wang et al. 2007). The question for designers and investors is: when can information sharing between semantic applications be considered to be successful or not?

The driving force behind developing semantic e-government applications is the expected added value through enhanced semantic interoperability which means that the

M.A. Wimmer, H.J. Scholl, and E. Ferro (Eds.): EGOV 2008, LNCS 5184, pp. 291–302, 2008.

“precise meaning of exchanged information is understandable by any other application that was not initially developed for this purpose” (European Communities, 2004, p.16). Here, “understandable” does not necessarily refer to machine intelligence but to the ability to combine received information with other information resources and to process it in a way that the meaning it has for the service users (citizens, businesses, even other administrations) and/or for the service providers is sustained. However, the exchange of information implied by semantic interoperability spans much more than just technical components working together effectively without prior communication. In many cases, the result of one semantic application is an informational resource for another, and the information exchange mostly appears to be unidirectional, deferrable, and/or very loosely coupled. Furthermore, the information is intended to be shared within an open world with distributed applications, multiple stakeholders (providing and consuming information), heterogeneous data, diverse ownerships and a never ending stream of new information being added.

Certainly, only that can be improved for which we have definite objectives *and* a method how to measure to what extent we have reached our goals. However, in open information sharing environments as described above (of which the Semantic Web is the most prominent instance) existing approaches focusing on information sharing success are not sufficient because there is no closed system and we cannot even assume a coherent social context as a source of reliable semantic integrity constraints (in contrast to e.g. assuring semantic integrity of a corporate ERP system). Therefore, this paper concentrates on testing strategies that have the potential to systematically guide development efforts towards improving semantic e-government applications aiming at successful information sharing. The underlying research question is: what are test objectives and test instruments which may inform stakeholders effectively about progress in developing semantic e-government applications?

The research presented in this paper is induced by the need for systematic testing within the Access-eGov research project (see access-egov.org). However, the approach to answer the above research question goes beyond this specific project: we develop a framework for identifying test focus and test objectives based on the assumption that automatic information processing based on encoded meaning is the core of semantic e-government applications to be evaluated. Taking into account test strategies from software engineering and IT project management as well as different stakeholder perspectives, we identify possible test approaches. Several of these approaches have been applied in the Access-eGov project and we discuss the experience gathered in view of the newly developed framework in order to identify lessons learnt as well as to point to future research.

The contribution of the paper is a portfolio of strategies testing specifically the value added by semantic technology. Based on the distinction of systems view, agent view, and user view, we conclude that in order to improve semantic e-government applications the evaluation should be carried out from all three of these views and we suggest certain test instruments for each view. Since there are hardly any instruments for testing if the data and its encoded meaning in focus are successfully processed by other applications (agent view) we suggest future research to focus on applying a test-first approach, e.g. through programming a test agent upfront, in order to improve the technical design and implementation as well as the overall performance in terms of semantic information processing.

The structure of the paper follows the research approach as outlined above: in the next section we develop a framework to identify test focus and test objectives through following up the automatic information processing based on encoded meaning. The third section reviews the state of the art of testing in application development as far as relevant for identifying test objectives and appropriate test instruments. The forth section reports on test approaches and lessons learnt within the Access-eGov project. The final section summarizes the findings through relating the strengths of various test instruments to the three different views and proposes future research.

2 Automatic Information Processing Based on Encoded Meaning

Within this article, we consider e-government applications from the point of view of information management and information processing: We seek to trace how—on top of providing existing e-government information and services—new semantic technology contributes to combining received information with other information resources and to processing it in a way that the meaning it has for the application users (information providers and consumers) is sustained. To this end, we develop a framework to identify (a) steps and functional components of automatic information processing based on encoded meaning, and (b) test objectives as well as suitable points of process interception for test purposes.

Ad (a): for identifying steps and functional components we take advantage of semantic technology enabling data involved in every process step to be automatically linked to the relevant context of its interpretation. Hence, in a semantic e-government application we can *observe*, i.e. test, the following by viewing the application from inside and/or taking the view of a user or machine agent interacting with the application:

1. *System view:* Information is being processed from sources to targets while in every major process step the data in focus is successfully linked to the encoded meaning of the data, i.e. to relevant interpretation context.
2. *Agent view:* The “precise meaning of exchanged information is understandable by any other application that was not initially developed for this purpose” (EC 2004, p.16); prerequisite for this is the provision of formats for integration and combination of data drawn from diverse sources as well as a language for recording how the data relates to real world objects (cf. www.w3.org/2001/sw/).
3. *User view:* The targeted information consumers find the data provided by the application valuable according to their own interpretation context and/or accept the interpretation context intended to be shared by the information sources (administrations, for most parts). According to their roles as application users (e.g. citizen, business representative, administrative employee), they usually have different intentions and ways of interpreting the same data or so-called boundary objects (Klischewski & Ukena 2008).

The idea of an information process and the basic functional components enabling a semantic e-government application are depicted in figure 1. It shows the two groups of users involved in information sharing: The information providers to the left and the information consumers to the right. The information providers create service description and ontologies (used for describing services), thus encoding the information in

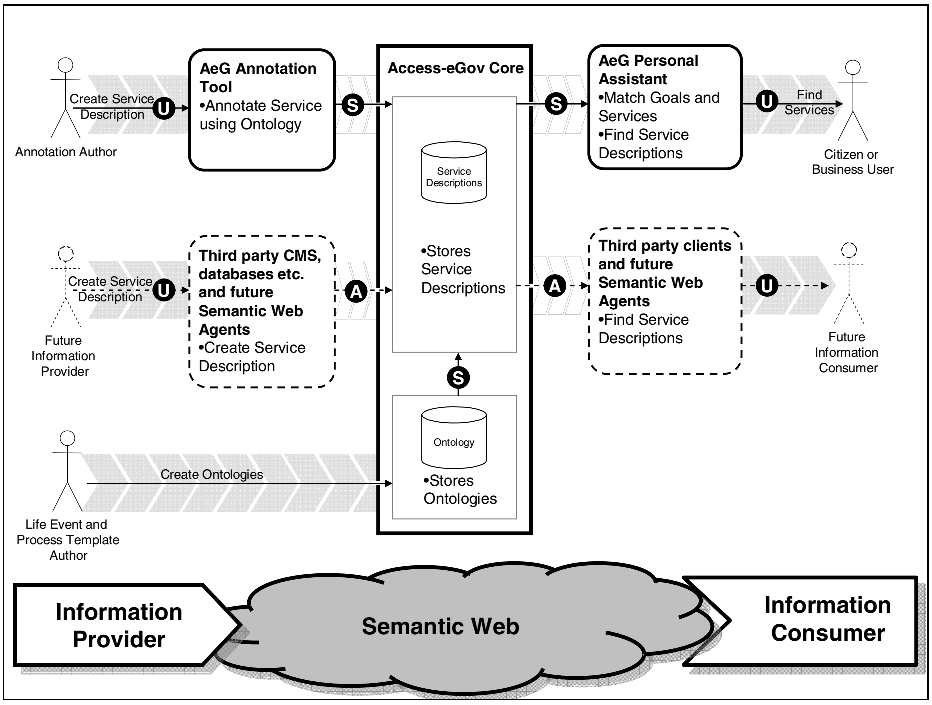

Fig. 1. Information flows from information providers (left) through a semantic infrastructure (middle) to information consumers (right)

such a way that the information's context is formalized. This enables Semantic Web applications, like the Access-eGov Core, to process the information, thus making it available to information consumers. The information consumers try to contextualize the information with whatever formalized context may be made available to them through the user interface. Information providers may either use the Annotation Tool provided by Access-eGov (top left corner) or they may use third party CMS's, databases or software agents (middle left) that are adopted to provide information to the Access-eGov core in different ways (e.g. a web-service API or by semantically annotated content). Information consumers can use the Access-eGov Personal Assistant (top right) to find services. In the future they may also use other clients or Semantic Web agents developed by third parties (middle right).

Ad (b): in order to observe if the data in focus is successfully linked to its relevant interpretation context, we seek to identify specific test objectives as well as suitable points of process interception for the purpose of testing. In figure 1 candidate interception points for testing are marked with a circled "S", "A" and "U". The above framework alerts us to three basic types of tests questions:

1. Is the data in focus successfully linked to its relevant interpretation context *represented within the machine environment*? (Interception points marked "S")
2. Is the data and its encoded meaning in focus successfully processed ("understood") *by another application, i.e. agent*? (Interception points marked "A")

3. Is the data in focus successfully linked to its relevant interpretation context *by the application user*? (Interception points marked "U")

All three of these types of interceptions require different approaches to actually test if the objectives of the semantic application are met.

3 Test Approaches in Application Development

This section reviews the state of the art of testing in systems development as far as is relevant for identifying test objectives and appropriate test instruments.

Test instruments are commonly characterized along two dimensions: (1) being either static (i.e. tests that do not require the code to be executed) or dynamic (i.e. tests that require code execution), or (2) with respect to the level of knowledge that is necessary to design the test for a given system, i.e. black box testing (requiring no knowledge of system internals), grey box testing (requiring some knowledge of system internals), or white box testing (requiring full knowledge of system internals). Different quality attributes and structural levels may be tested by using different kinds of tests. While there is no one-to-one relation between test objectives and test instruments, some instruments are better suited than others for testing certain qualities or levels.

Software testing may be defined as "[…] a process used for revealing defects in software, and for establishing that the software has attained a specified degree of quality with respect to selected attributes." (Burnstein 2003, p. 7) This broad definition illustrates the dual nature of testing, i.e. testing to discover defects and deviations from the requirements (verification) and testing to evaluate quality attributes like usability and reliability (validation). Common quality attributes are correctness with respect to given specification; reliability with respect to required functions under specified conditions; usability with respect to efficiency of use, learnability etc., and interoperability, i.e. the ability of systems and/or components to exchange data with each other (cf. Burnstein 2003). These objectives can be related to the views defined in the previous section.

Systems view: For trivial applications systems, testing may be possible for the system as a whole. For more complex systems it is common practice to divide the testing effort into different levels which correspond to the structural levels of the software:

- Unit testing, which aims at testing individual parts of a software, like single methods of a class
- Integration testing, which tests several components at the same time
- System testing, which tests the system as a whole (acceptance testing also falls into this category)

Depending on the software development methodology that is used, testing may be viewed as a one-time effort or a continuous process during the life-cycle of the software. For example, in the classic Waterfall model of software development, the testing phase directly follows the coding phase and precedes the operation phase (cf. Royce 1970). With the advent of iterative development methodologies testing becomes a continuous effort throughout the development process (cf. e.g. the Testing

Maturity Model in Burnstein 2003) or even the driving force of the software design as in Test-Driven-Development (e.g. Beck 2002).

Besides distinguishing testing approaches by their temporal appearance in the development cycle, testing may also be categorized by the test objectives, the tested object or level, and the test instrument.

With the advent of agile software development methods that require short iterations and continuous integration, tests became more and more important. Tests do no longer only serve as means of validation and verification but can be an integral part of the development cycle to support the re-factoring of code by ensuring that the re-factored code passes all available tests. "Test-first" approaches suggest starting the implementation of a piece of software by first implementing tests for each of the required functionalities. Advocates of test-driven-development suggest that the main driving force of the software design should come from these tests.

Agent view: For testing semantic applications from the point of view of its expected unique contribution (i.e. whether the data and its encoded meaning in focus is successfully processed ("understood") by another application—see section 2) no systematic approaches have been published yet. Extant literature or other sources that could be related to "semantic (application) testing" focus mainly on testing ontologies and/or automatic reasoning using ontologies. Examples of the latter are the provision of Web Ontology Language test cases (www.w3.org/TR/owl-test/) or discussion of complexity in subsumption of clauses in the language of first order logic (Marcinkowski et al. 2005). With the aim of eventually providing a test bed, Aleman-Meza et al. (2004) have proposed an ontology (SWETO) that incorporates instances extracted from heterogeneous sources for testing purposes.

The capabilities of information processing can be partly tested by so-called "competency questions" which have been suggested and used to guide the development of ontologies as well as to test ontologies (e.g. Gruninger and Fox 1994, Staab et al. 2001). Competency questions are first informally stated in order to guide the design of an ontology. A competency question usually poses a simple or complex question that the future ontology should be able "to answer". At a later stage, the competency questions are often formalized to serve as automated tests for the ontology. However, despite much attention on Semantic Web applications, systematic testing from an agent perspective is still a field waiting for much research to be done.

User view: Different stakeholders usually have different objectives with respect to the purpose of the system under development. This results in different objectives with respect to testing the application in focus. In addition to application users we find "non-users" such as application sponsors, information managers and system developers, each with another distinct view on the information processing in focus. For example, on the one hand the head of administration (a sponsor) might aim for savings and improvements resulting from efficiency gains and increased interoperability, while trying to keep the required effort to reach these benefits as low as possible. Thus, she will want to evaluate the system in this regard. On the other hand, developers may want to test in such a way as to guide the enhancement of semantic structures that fulfil the requirements of their customers. In particular, they want to know how to identify the relevant concepts in the domain of interest, how to limit the scope of this analysis effort, and how to get all relevant information they need for implementation.

Within the testing framework (cf. figure 1) we focus on those stakeholders who participate in the transfer of information, i.e. who will genuinely use the system either as information providers or as information consumers. Possible test instruments are:

- think aloud sessions: a user is asked to perform a certain task with the help of the newly developed system (or system under development). The user is instructed to "think aloud", i.e. to say out loud what is going on in her mind while using the system. The whole session is usually video-taped for later analysis by an expert. (Cf. Joergensen 1989)
- workshop: several users are asked to use the system, discuss their experience and provide feedback for improvement
- questionnaire: a questionnaire with closed questions that assess information quality and usability aspects of an application from the user's point of view using answers on a predefined scale allows comparison of results across different regions and pilots (cf. Elling et al 2007)

4 Testing Semantic Information Processing in Practice: Experience from a Field Test

This section reports on test approaches, experience, and lessons learnt within the Access-eGov project. The trial in Schleswig-Holstein involved several administrations in different municipalities and communities whose officers had to annotate a number of services related to marriage. The officers used a software component called Annotation Tool (AT) to create these service annotations. Prospective users of the marriage-related services (i.e. citizens) were asked to use another software component, the Personal Assistant Client (PAC), to look up relevant service information.

The project's evaluation strategy was first outlined in project deliverable D2.2 "User requirement analysis & development/test recommendations"[1] and defined in D8.1 "User requirement analysis & development/test recommendations". The main focus of the evaluation strategy lay on the information quality criteria from the information consumer's point of view as it is described by Lillrank (2003). These information quality criteria were the basis for most testing efforts of the first trial. In addition, systems developers introduced their own criteria for testing.

Developer testing: The developers relied on component testing and integration testing using the well known JUnit-framework. The component tests were developed as *ex post* white box tests, i.e. the tests were developed after the components which were to be tested. Most tests were written for complex components, which usually comprise several classes. Integration testing focused on core functionalities that are provided by different sets of components. In addition to functional tests, the developers performed performance and scalability testing for the most critical components.

Think-aloud: The think-aloud sessions were conducted with citizens (using the PAC) and an administration officer (using the AT). Video-taped think-aloud sessions yield rich qualitative data that can reveal many problems users may have in using a system. In our case they revealed difficulties that information providers had when creating the

[1] All public deliverables are available at: http://www.accessegov.org/acegov/web/uk

service description (e.g. missing data fields or insufficient description of the intended content of a field). They also revealed where citizens had problems interpreting the information in the PAC. The four citizens, who participated in one session each, were between 20 and 40 years old with ages roughly evenly distributed across this range. They were recruited among colleagues, friends, and friends of colleagues based on the fact that they either planned to marry in the near future or had married in the recent past. The think-aloud sessions with the citizens were conducted at the premise of the Schleswig-Holstein user partner by one of the user partners. Sessions lasted between 15 and 45 minutes, depending on how thoroughly the citizens decided to read the provided information. All citizens were asked to perform the same tasks related to marriage services in Schleswig-Holstein as they would normally do or had done.

The administration officer was around 40 years of age working part-time as the public relations officer being also responsible for the communal web-site of the community with around 200,000 residents. This think-aloud session was conducted by one of the authors at the officer's office. The officer was asked to annotate different services by using the annotation tool. The session lasted about 45 minutes.

Workshop: One half-day workshops were conducted with the officers responsible for annotating the services. This workshop had two goals: 1) to introduce the officers to the annotation tool prototype and its usage, and 2) to collect feedback on any issues with this prototype. The workshop was video-taped for later analysis.

The 13 participants were either registrars responsible for performing marriage related services and marriages or Internet editors responsible for authoring communal web-sites. During the workshop the officers used the Annotation Tool to describe some of the marriage-related services their administration provides. After a short period of getting acquainted with the system, the officers provided feedback on their experience. One large concern of the officers was that the user interface did not provide sufficient context which in some cases made it difficult for them to know what kind of data should be entered and in what way it should be structured. For example, if one creates a link to another site using the appropriate option in the Annotation Tool, what should one enter in the provided fields labeled "URL", "Name", "Description"? However, while the officers called for "more context", they were still able to describe the services as intended.

Online questionnaire: In order to collect quantitative data that can be compared across regions and trials, we extended an available questionnaire for evaluation of web-sites (Elling et al 2007). Most questions aim at the mentioned information quality and usability criteria and must be answered on a given scale. The questionnaire was distributed as an online-questionnaire. The URL was published through an announcement of the states press office inviting the public to try the Personal Assistant and to answer the questionnaire afterwards. In addition, participating officers were asked to inform citizens about this online service during their regular work activities. The online questionnaire was available for a period of four weeks and was completed 69 times (incomplete submissions: 223).

Lab test: In addition to real-life testing by citizens we also conducted a series of test sessions that took place at a test lab at the German University in Cairo. The lab test was designed to test PAC, i.e. the information consumer perspective. Testers were recruited among the students and were asked to perform and document a series of tasks. The results were evaluated with regard to completeness and correctness of the retrieved information. The test lab did not perform any technical tests, like load testing. Instead, the tasks were designed a way that the testers could easily identify with it, which was intended to bring the test as close to real-life situations as possible. Each tester had to perform the same series of three different tasks during three test sessions. 14 testers performed a total 42 test tasks over a period of three days. Each task took around 90 minutes to complete (including instructions etc.). One task focused on those aspects of the Personal Assistant Client that deliver non-personalized information. The other two tasks focused on the personalized information.

Lessons learnt: Though the original intention of *think-aloud sessions* is the testing of usability issues, they are also suited to reveal users' problems with respect to information processing, both on the information consumer as well the information provider side. However, think-aloud sessions are resource intensive tools that cannot easily be used on a larger scale. The *workshop* showed that information providers want to know what the information will look like when it is presented to the information consumer. However, it also showed that information provision is possible without this kind of support. This is particularly important because it is counter-productive to any future application if information providers tailor their information too specifically for any particular information consumer agent and/or user interface. The *questionnaires* are well suited to survey a large sample of information consumers and yield comparable results across different regions, but they can only serve as a coarse tool for identifying issues that may—or may not—be caused by missing contextual information. Thus, questionnaires should always be followed up by more thorough investigations with other instruments. Though the questionnaire itself is a coarse tool that does not allow the identification of specific problems, it is helpful to provide an overview of how the system's information quality is perceived by a larger number of users. The information qualities that are rated low can then be investigated in more detail by using more specific instruments that yield richer results. The *lab test* surfaced some bugs in the software and also pointed to some usability issues that later were confirmed by the think-aloud sessions. More importantly, it showed that the majority of the testers were able to successfully complete the given tasks and interpret the information as intended. For example, testers were generally able to find the address of the responsible office for a certain service. This indicates that the Personal Assistant Client provides information in a manner useful for citizens.

All in all, the project's evaluation strategy was quite successful on the third testing question (cf. section 2) but could benefit from some improvements: developers testing should be more aligned with field test instruments in order to integrate the results. In particular, all the employed test instruments in the field test relied on human testers and user-interface testing. Therefore it needs additional instruments testing the information processing focusing on the agent view (thus bypassing the additional impact of the application interfaces on information processing).

5 Discussion: Software Agents for a Test-First Approach?

In this paper we set out to identify suitable test instruments helping to answer the three types of test questions according to the three views introduced in section 2.

Based on our theoretical considerations (section 2 and 3) as well as our practical experience (section 4) we can conclude recommendations for testing semantic applications and point to future research.

As all three views (system, agent, user) are relevant, we recommend that any test strategy for improving semantic e-government applications should implement test instruments for all three of these views. In application development a variety of test instruments are applied, each with different deliverables. Table 1 summarizes our findings pointing mainly to the strengths of each instrument, supported by our experience in the Access-eGov project (the recommendation could be extended to a full SWOT analysis). The check marks ✓ or (✓) in any table cell indicate that this test instrument is considered useful or partly useful to answer test questions in relation to the system, agent, or user view.

Table 1. Test strategy portfolio

Test instrument	System view	Agent view	User view
Developer testing	✓ Timely feedback on system performance	(✓) timely feedback if suitable test agent available	
Test lab	(✓) Independent, but not timely feedback	✓ Variety of test opportunities	(✓) Feedback from controlled but "artificial" environment
Think-aloud			✓ Rich data about information use and system interaction
Workshop			✓ Comprehensive understanding of stakeholder views
Question-naire			✓ Large scale feedback on user satisfaction

According to this table the test lab has the potential to cover testing from all views, at least to some extent. However, instruments for semantic test labs have not yet been developed or proved to be effective means—here we see the most pressing need for future research of testing semantic applications.

In case of the Access-eGov project, the Personal Assistant Client provides additional contextual information that does not originate from a single information provider. Evaluation has shown that information processing is possible using the developed components (Annotation Tool, Personal Assistant Client and Core), but we have not shown that information sharing is indeed possible in conjunction with other agents. To this end, we suggest the use of "test agents", i.e. software agents that can be used to provide information to and/or retrieve information from the semantic application in focus by using common-place means e.g. a Web-service application programming interface. Such test agents should be able to execute different test scenarios, which are designed to resemble real-life use cases from a technical point of view.

For example, in the Access-eGov project, one scenario would deal with providing e-government services descriptions based on external data sources; and another scenario would test if these descriptions can be retrieved and processed correctly when combined with other relevant information.

Taking into account the broad experience of software testing as well as our own project experience, we conclude that improving semantic e-government applications can be effectively supported through applying a test-first approach, e.g. through providing e-government test agents to be used in test labs or directly within the development process. However, more research is needed to identify what kind of scenarios and functionality such test agents exactly should perform in the area of e-government. These scenarios must be defined from the e-government information management perspective: needing a selection of use cases that resemble typical e-government user interaction, and the information processing should make reference to pools of relevant and accessible data (e.g. resources on e-government websites) as well as to agreed standards of information structures (e.g. e-government service directories) and already standardized and encoded meanings (e.g. ontologies).

Such test agents—if they are fairly easy to handle, are able to technically connect and provide a variety of scenarios for execution—can ensure that any semantic application under development could be sufficiently challenged to test whether the data in focus is linked to its context represented, whether it can be "understood" by another application and/or whether application users actually can get the information they are looking for. Furthermore, standardization of an e-government test agent itself has the potential for significantly raising the level of semantic interoperability: if widely applied, such an e-government test agent could ensure that all semantic e-government applications (e.g. within a certain region) will meet the same performance criteria and will be able to automatically connect their information processes. This, indeed, would be a great step forward to implement the vision of Semantic Web for e-government.

Acknowledgements. This research has been partly funded by the European Commission under the project no. FP6-2004-27020. Thanks to all Access-eGov colleagues for giving us the testing opportunities. Furthermore, we are especially thankful for a number of reviewer comments which helped improving the paper.

References

1. Abecker, A., Sheth, A., Mentzas, G., Stojanovich, L. (eds.): Proceedings AAAI Spring Symposium "Semantic Web Meets eGovernment", Stanford University, March 27-29, 2006. AAAI Press, Menlo Park (2006) Technical Report SS-06-06
2. Beck, K.: Test-Driven-Development. Pearson (2002)
3. Beust, C., Suleiman, H.: Next Generation Java Testing: TestNG and Advanced Concepts. Addison-Wesley, Reading (2007)
4. Aleman-Meza, B., Halaschek-Wiener, C., Sheth, A.P., Arpinar, I.B., Sannapareddy, G.: SWETO: Large-Scale Semantic Web Test-bed. In: Proceedings of the 16th International Conference on Software Engineering & Knowledge Enginerring (SEKE 2004): Workshop on Ontology in Action, Banff, Canada, June 21-24, 2004, pp. 490–493 (2004)
5. Burnstein, I.: Practical Software Testing. Springer, Heidelberg (2003)

6. Elling, S., Lentz, L., de Jong, M.: Website Evaluation Questionnaire: Development of a Research-Based Tool for Evaluating Informational Websites. In: Proceedings of the 6th International EGOV Conference, pp. 293–304. Springer, Heidelberg (2007)
7. EC European Communities (ed.): European Interoperability Framework for pan-European eGovernment Services (version 1.0). Office for Official Publications of the European Communities, Luxembourg, (2004), http://europa.eu.int/idabc/en/document/3761
8. Gruninger, M., Fox, M.S.: The role of competency questions in enterprise engineering. In: Proceedings of the IFIP WG5.7 Workshop on Benchmarking-Theory and Practice, Trondheim, Norway (1994)
9. Furdik, K., Hreno, J., Sabol, T.: Conceptualisation and Semantic Annotation of eGovernment Services in WSMO. In: Snášel, V. (ed.) Proceedings of the conference Znalosti (Knowledge), Bratislava, Slovakia, February 13-15, 2008, pp. 66–77 (2008)
10. Furdik, K., Sabol, T., Bednar, P.: Framework for integration of e-Government Services on a Semantic Basis. In: Electronic Government, 6th International EGOV Conference. Proceedings of ongoing research, project contributions and workshops, Regensburg, Germany, September 3-6, 2007, pp. 71–78. Trauner Verlag, Linz (2007)
11. Joergensen, A.H.: Using the "thinking-aloud" method in system development. In: Salvendy, G., Smith, M.J. (eds.) Designing and using human-computer interfaces and knowledge-based systems, pp. 743–750. Elsevier, Amsterdam (1989)
12. Klischewski, R., Ukena, S.: Designing semantic e-Government services driven by user requirements. In: Electronic Government, 6th International EGOV Conference. Proceedings of ongoing research, project contributions and workshops, Regensburg, Germany, September 3-6, 2007, pp. 133–140. Trauner Verlag, Linz (2007)
13. Klischewski, R., Ukena, S.: An Activity-based Approach Towards Development and Use of E-government Service Ontologies. In: Proceedings HICSS-41. IEEE, Los Alamitos (2008)
14. Lillrank, P.: The quality of information. International Journal of Quality & Reliability Management 20(6), 691–703 (2003)
15. Marcinkowski, J., Otop, J., Stelmaszek, G.: On a Semantic Subsumption Test. In: Baader, F., Voronkov, A. (eds.) LPAR 2004. LNCS (LNAI), vol. 3452, pp. 142–153. Springer, Heidelberg (2005)
16. Royce, W.: Managing the Development of Large Software Systems. In: Proceedings of IEEE WESCON, vol. 26, pp. 1–9 (1970)
17. Scholl, J., Klischewski, R.: E-Government Integration and Interoperability: Framing the Research Agenda. International Journal of Public Administration (IJPA) 30(8-9), 889–920 (2007)
18. Skokan, M., Bednar, P.: Semantic orchestration of services in eGovernment. In: Snášel, V. (ed.) Proceedings of the Czech-Slovak conference Znalosti (Knowledge), Bratislava, Slovakia, February 13-15, 2008, pp. 215–223 (2008)
19. Staab, S., Studer, R., Schnurr, H.-P., Sure, Y.: Knowledge processes and ontologies. In: Intelligent Systems, vol. 16 (1), pp. 26–34. IEEE, Los Alamitos (2001)