The initial description of the project (February 2010)

Title: **TESTING AND FORMAL MODELING OF SERVICE-BASED SOFTWARE APPLICATIONS**

Principal Investigator: **Dr. Alin Stefanescu**

Abstract:
Software testing is a very expensive, tedious and time consuming task and it is estimated to require up to 50% of the total cost of software development. This cost could be reduced if the process of testing is automated. Therefore, systems that can automatically generate test cases are studied by many researchers and practitioners. One of the most attractive and recent approaches for test design automation is model-based testing (MBT), which uses formal models describing the system under test for automatic test case generation. While the theory of MBT was mainly developed and applied to complement the software engineering area, there is a new paradigm gaining a large and quick popularity and industrial adoption, which is the service-oriented architectures (SOA), identified also by the EU FP7 as one of the areas with highest European innovation potential within the vision of the future Internet of Services. The advantages of operability, flexibility, adaptability, and functionality promised by SOA come together with several challenges to be addressed regarding their robustness, dependability and quality. The aim of this project is to advance the state of the art and propose new solutions for the quality assurance of service-based systems using MBT technologies. First, the project will study the different model types that are suitable for the area of service integration testing. Second, it will address the challenges of test data generation using an original combination of MBT with biology-inspired evolutionary techniques and metaheuristic search algorithms. Finally, it will try to address the scalability problems usually faced by practitioners, by keeping a close contact with large SOA-enabled software providers like SAP and validating the research results on the provided industrial benchmarks.

The main goal:
This research project will investigate new and efficient methods for the model-based testing of service-based software applications using evolutionary techniques. The main ingredients are found in the three areas: (a) formal test modeling for service oriented architectures, (b) automatic generation of test cases and test data, and (c) the application of the research results to state-of-the-art and state-of-the-practice use cases. The goal is to devise methods that have not only clear theoretical advantages over the current methods in the research literature, but also scale on real-life examples from industry.

The importance and the relevance of the scientific content:
Service-oriented architecture (SOA) is a recent approach from the area of distributed systems where the software services are the core entities in the structure and interoperability of the system. SOAs are becoming mainstream in the software industry as presented by a Forrester survey [1] that states that 67% of companies taking part in the survey (i.e., 2200 across North-America and Europe) expect to be using SOA by the end of 2009, while 60% of those currently using it are expanding their usage. This trend will certainly continue towards the envisaged Internet of Services (IoS), in which services will be

globally available through the Internet. Service-based systems (SBSs) built from those IoS services, will provide significant advantages (over "in-house" services) in terms of operability, flexibility, adaptability, and functionality. However, due to the distributed nature of these services, and the lack of control and knowledge on them, building SBSs on top of IoS services will require addressing key research challenges to engineer robust, dependable and high quality SBSs, especially in the area of SOA testing – see [2,3,6,7].

Model-based testing (MBT) is a class of techniques to automatically generate tests from a formally-specified model describing the system under test, thus increasing the efficiency and coverage of the testing process. The MBT paradigm recently received a lot of attention both in the academic as well as the industrial communities, with several books published [11, 12, 13] and several dedicated workshops (A-MOST, MBT, MOTES, MOTIP, etc.). Moreover, there is already a market with several MBT commercial tools (like Qtronic from Conformiq, TestDesigner from Smartesting, and SpecExplorer from Microsoft). However, most of the tools and methods work for functional testing during the software development phase and few methods are available for the SOA paradigm. More precisely, we could identify four different levels of SOA testing: unit testing (the testing of service operations and internal classes), single service testing (testing the integration of the functional units inside a single service), integration testing (testing that several service components work well together) and system testing (testing the system as a whole). Until now the focus was on the first two types rather than the latter two ones, that is, the improving the test automation in the domain of service developers and service providers prevailed over the domain of service integrators or service consumers.

After the succinct introduction of the SOA and MBT contexts, let us understand the status of the field and emphasize research and industrial challenges still lying ahead of us. Such challenges and research opportunities can be identified in each of the main steps of the application of MBT to service-based systems. These steps are: (a) the definition and provision of formal test models, (b) methods for test case and test data generation based on these models, and (c) the application or concretization of the generated tests onto the service-based system under test. We describe below each of these three areas separately:

(a) Formal modeling for SOA testing

As of today, many different model-driven techniques for quality assurance (like testing, monitoring, formal verification) of SBSs have been proposed as examined in [6] and [8]. However, most of these techniques rely on ad-hoc models and languages that are tailored to the specific class of properties to be tested and to the specific tool: for example, the 30 testing approaches analyzed in [8] use 16 different models and languages.

That means there is not yet a de-facto standard for SOA testing. This is on the one hand due to the fact that the SOA modeling stack need to cover several layers, from single service descriptions to complex service compositions and on the other hand that the research for MBT for SOA is still in a consolidation process influenced by the fast developments in the service engineering area. However, we can identify two groups of modeling languages for SOA: UML-based and WS-* based. The UML standard (see http://www.uml.org) is the de-facto modeling language for classical software engineering and there are a couple of profiles for UML that allow for testing modeling, like the U2TP standard (see http://utp.omg.org) and also recent UML profiles for SOA (like SoaML and UML4SOA) profiles. However, there is no current research that looks at a combination of the two profiles that would enable a MBT based approach for SOA based on UML standards. Moreover, UML needs in many cases a more precise semantics in order to be used for MBT. On the other hand, service engineering standards are based on Web Services (WS) languages (see http://www.w3.org/standards) and provide different types of language like WSDL, WS-BPEL, WS-CDL for service descriptions. However, although there is MBT research taking into account models like WSDL and WS-BPEL describing single services or service orchestrations [6], there is not enough research in the area of testing for service choreographies (different from orchestrations, choreographies assume a distributed control rather than a centralized control of service communication). Such choreographies could provide a very good basis for service integration testing [15]. Moreover, new standards like BPMN 2.0 supporting choreography modeling were not yet investigated from the MBT point of view. Last but not least, there are also not many models ready to use for system testing. Such testing needs to work on a level of Graphical User Interface (GUI) for service compositions and MBT for GUI is still a recent area of research [10], especially for service-based applications [16].

(b) Test generation and test execution for SOA systems

Given a formal test model, the next MBT step is to automatically generate test cases (i.e., paths to be executed on the system under test) according to certain coverage criteria. Very powerful test generation methods, especially for state-based models, have been devised in the literature [11]. However, trying to apply them to service-based systems, we can identify at least three important drawbacks:

- Test data: Given their loose coupling, distributivity, and heterogeneity, the service-based systems rely in in many cases on asynchronous communication using complex XML-like data. These data usually exhibit intricate internal dependencies as well as references to existing master data and the actual system configurations [17]. This makes the test data provision for such systems a highly intensive activity for testers and increases the difficulty of automatic test data generation. There exist many test data generation methods in the literature [19], based on e.g. constraint solving, symbolic execution, domain reduction, or metaheuristic search, but they were mostly focusing on white-box testing, i.e. the tester had access to the code of the tested application. This is in most cases not the case for service-based applications where only the service interfaces and the communication protocols are exposed.
- Integration and system testing: Secondly, most of the approaches in the service testing literature [6] concentrate on single service testing rather than integration and system testing. Under the assumption of hard-coded service integration, existing techniques for integration testing of component-based systems could be applicable. These are mainly using UML-like static and behavioral models (class diagrams, state machines, collaboration, and interaction diagrams) to generate tests. However, the assumption does not hold in IoS as service compositions are loosely coupled. Moreover, many of SOA models used

in practice come from the realm of WS-standards rather than UML and information necessary for test generation is missing from the metamodels.

- Design vs. runtime testing: Finally, given the dynamic and adaptable nature of SBSs, many of the traditional MBT techniques that involve only test generation at design time might not be very efficient to discover integration issues that occur at runtime or during dynamic system reconfiguration, as discussed in [7]. Therefore, there is a clear need of new MBT techniques adapted for runtime testing explicitly targeting such faults during the service discovery, service publishing, service composition and service binding activities at runtime.

(c) Industrial application of MBT research methods

Additionally to the research challenges identified above, there is also a stringent need for improvement also in the industry. According to Gartner [9], the state-of-the-practice is quite far from being able to handle the quality challenges in the IoS setting, as the leading quality assurance tools in the emerging SOA market (i.e., Green Hat, Parasoft, Soasta, Progress Software, iTKO) concentrate on testing and monitoring single services rather than complex service compositions, and on static architectures rather than highly dynamic ones. One reason for this is that information (like formal descriptions of service compositions and complex consistent test data) that could be used to automate this process is currently missing, both during design- and run-time. This is also indicated by industry surveys [4] and [5] showing that the majority of SOA and software projects suffer from insufficient and inadequate testing. Another reason for this problematic situation is the lack of sound and comprehensive techniques and tools, with the consequence that many of the surveyed IT managers have decided to implement in-house and ad-hoc testing processes. Moreover, there is a proliferation of domain-specific languages (DSLs) used in industry that are very benefic in their specific context by high specialization and dedicated application area, but have the clear drawback that standard MBT techniques cannot be easily applied due to the different peculiarities of the DSLs that must to be taken into account. For example, SAP, the leading software provider of business applications and a promoter of Enterprise SOA [18] uses more than 13 proprietary model types for their SOA architecture. Last but not least, as mentioned in [11], in practice the concretization of the abstract test cases generated using MBT (i.e. making the test cases executable by providing the manual test data or setting up the test configurations) can require up to the same effort as for the modeling and test generation steps.

Evolutionary Testing (ET) is a particular case of a search-based approach to test data generation. Search-based techniques (like genetic and evolutionary algorithms) are designed to find good approximations to the optimal solution in large complex search spaces, especially in the difficult cases when formal analytical and classical test generation methods fail because of the combinatorial explosion of possible interleavings or test data combination. The search is driven by so-called fitness functions that are used to optimize the test data towards a failure of the system behavior. Evolutionary techniques apply algorithms inspired by biological evolution such as reproduction, mutation, recombination, natural selection, and survival of the fittest in order to find proper test data for the optimization of the fitness functions. Until now, evolutionary testing concentrated on the application of such techniques in white-box (structural) testing, as seen in the survey from [20]. The application of search-based techniques in the area of black-box or model-based testing is a far less frequent topic [21]. While a promising technique that attracted also the EU funding in the project EvoTest (EvoTest – Evolutionary Testing for Complex Systems – 2006-2009), the application of evolutionary testing to SOA testing was not yet considered. We could only identify a couple of very recent papers that use search-based techniques and even there the focus was on service level agreements [22] and orchestrations [23]. So this is a wide unexplored area that we investigate in our project.

Interdisciplinarity of the proposed project

This project will provide a strong link between two distinct areas of computer science: theoretical computer science (listed as CNCSIS-UEFISCU priority domain 1.5 under Natural Sciences - see Annex 2) and system and software engineering (listed as CNCSIS-UEFISCU priority domain 2.2 under Engineering Sciences – see Annex 2). The formal modelling and model based testing belong to the former area, whereas the service-based systems belong to the latter, the synergies of the two areas being only very recently more carefully studied. In addition, the obtained results will be applied in multidisciplinary case studies, to be performed in collaboration with researchers from SAP, concerning modelling and testing of enterprise service systems. By multidisciplinary use cases we mean for instance use cases from the area of Internet of Things [24], where dynamic networks of objects (cars, mobile phone, smart home appliances, etc.) are communicating using service-oriented architectures. Finally, on a technical level, there is another high interdisciplinarity in the use of evolutionary algorithms (inspired from biology) and their application to SOA testing, as described in the following section.

Bibliography:

[1] Forrester Research, “Enterprise and SMB software survey”, North America and Europe, Q4 2008. Survey, 2008.

[2] Aberdeen Group, “SOA and Web Services Testing – How Different Can It Be?” Industry Survey. August 2007.

[3] D. Linthicum, and J. Murphy, “Key Strategies for SOA Testing”. Mindreef Inc., 2007.

[4] PR NEWSWIRE, “More than two-third (68%) of European IT Managers Believe That Products and Systems are Launched With Insufficient Testing,” Sep. 29, 2005. [Online]. Available: http://www.ulitzer.com/node/135839

[5] Green Hat, “Nine out of ten survey respondents agreed that inadequate testing is a major issue in delivering successful SOA projects using messaging middleware,” [Online]. Available: http://www.greenhat.com/news/newsTYIUSSSXRK.html

[6] G. Canfora, M. Di Penta, “Service-oriented architectures testing: A survey,” Software Engineering, Springer, pp. 78-105, 2009.

[7] M. Greiler, H.-G. Gross, K. A. Nasr, “Runtime Integration and Testing for Highly Dynamic Service Oriented ICT Solutions – An industry challenges report,” in TAICPART’09, IEEE Computer Society, 2009, pp. 51-55.

[8] S-Cube - the European Network of Excellence in Software Services and Systems. “Survey of Quality Related Aspects Relevant for Service-based

Applications". Deliverable PO-JRA-1.3.1, 2008. [Online]. Available: http://www.s-cube-network.eu/results/deliverables/wp-jra-1.3/PO-JRA-1.3.1-Survey-of-quality-related-aspects-relevant-for-SBAs.pdf

[9] Gartner, "Hype Cycle for Application Development, 2009". Gartner Research, July 2009.

[10] A. M. Memon, "An event-flow model of GUI-based applications for testing," Softw. Test., Verif. Reliab., vol. 17, no. 3, pp. 137–157, 2007.

[11] M. Utting, B. Legeard, "Practical model-based testing, a tools approach". Morgan Kaufmann, 2007.

[12] P. Baker, Z. R. Dai, J. J. Grabowski, Ø. Haugen, I. Schieferdecker, and C. Williams, "Model-Driven Testing: Using the UML Testing Profile". Springer, 2008.

[13] J. Jacky, M. Veanes, C. Campbell, W. Schulte, Model-based Software Testing and Analysis with C#. Cambridge University Press, 2008.

[14] M. Di Penta, G. Canfora, G. Esposito, V. Mazza and M. Bruno, "Search-based Testing of Service Level Agreements." In Proc. of the Conf. on Genetic and Evolutionary Computation (GECCO'07), pp. 1090–1097. ACM Press, 2007.

[15] A. Stefanescu, S. Wieczorek, A. Kirshin. "MBT4Chor: a model-based testing approach for service choreographies". In Proc. of 5th European Conf. on Model-Driven Architecture Foundations and Applications (ECMDA'09), LNCS, volume 5562, pp. 313-324. Springer, 2009.

[16] S. Wieczorek, A. Stefanescu. "Improving Testing of Enterprise Systems by Model-based Testing on Graphical User Interfaces". To appear in the Proc. of the 7th Workshop on System Testing and Validation (STV'10), IEEE Computer Society, 2010.

[17] S. Wieczorek, A. Stefanescu, I. Schieferdecker. "Test data provision for ERP systems." In Proc. of the International Conference on Software Testing, Verification, and Validation (ICST'08), pp. 396-403. IEEE Computer Society, 2008.

[18] D. Woods, T. Mattern, "Enterprise SOA – Designing IT for Business Innovation". O'Reilly, 2006.

[19] B. Korel, "Automated Software Test Data Generation." IEEE Trans. Software Eng. 16(8): 870-879, 1990.

[20] P. McMinn, "Search-based software test data generation: a survey." Softw. Test., Verif. Reliab., 14(2):105–156. Wiley, 2004.

[21] Raluca Lefticaru, Florentin Ipate, "Automatic State-Based Test Generation Using Genetic Algorithms". Proc. of SYNASC'07, pp. 188-195. IEEE Press, 2007.

[22] M. Di Penta, G. Canfora, G. Esposito, V. Mazza, M. Bruno, "Search-based testing of service level agreements". In Proc. of GECCO'07, pp. 1090–1097, ACM Press, 2007.

[23] R. Blanco, J. García-Fanjul and J. Tuya, "A first approach to test case generation for BPEL compositions of web services using Scatter Search". In Proc. of 2nd Int. Workshop on Search-Based Software Testing, pp. 131-140, IEEE Computer Society, 2009.

[24] Commission of the European Communities. "Internet of Things — An action plan for Europe". June 2009. Online at: http://ec.europa.eu/information_society/policy/rfid/documents/commiot2009.pdf

Objectives, methodology and expected results of the project:

The project has the main objective of providing an **integrated, end-to-end perspective for MBT for service integration**, starting with the test models based on the latest developments and models in the industry, continuing with new algorithms for MBT and ET optimized to these models, and finally validating their feasibility on real-life SOA scenarios. Given the timeframe of 24 months, the proposed project will not cover all the different challenges and improvement requirements identified in the previous section, but will concentrate on the topics where we expect to produce **a high research impact based on previous extensive experience of the applicant in the MBT for SOA field** [1,2,3,4,5]. The detailed objectives and the ideas of solving them are grouped according to the three areas identified in the previous subsections. Their time planning is provided in Section 9.1.

(a) Formal modeling for SOA testing

- **Objective a.1) Improvement of testability for SOA models.** Most of the existing work in MBT is based on classical UML or other state-based models, but there is still research needed to adapt those techniques to UML profiles for SOA (UML4SOA or SoaML) and combine them with existing UML profiles for testing (U2TP). As mentioned in the previous section, we are not aware of any work combining the two in an integrated framework. The applicant very recently started to investigate the applicability of the U2TP profile to service integration based on choreographies – see [1]. In this project, we will combine this work with the newly proposed UML profiles for SOA. Moreover, the applicant performed in [2] investigations on the necessary information that choreography models need to contain for an efficient MBT approach. We plan to use this knowledge to verify and increase the testability of existing choreography models like WS-CDL or BPEL4Chor. This will allow more intensive use of MBT for the choreography testing, thus complementing the prevailing body of research on service orchestrations.
- **Objective a.2) Advancement of modeling for GUI testing for SBSs.** Also recently the applicant investigated the testing at the GUI level for service-based applications [3]. Since there is interest from the industry and a research gap to be filled, this objective will investigate especially state-based models for GUIs that reuse testing meta-modeling elements.

(b) Test generation for SOA systems using evolutionary techniques

Since evolutionary and genetic algorithms by their nature involve adaptability and evolution, they are a good fit to the dynamic context of IoS where changes and reconfigurations under consistency assumptions happen at runtime. However, maybe due to the very recent and fast advancement of SOA, evolutionary testing (ET) was not yet used in this domain. As an innovative approach, we will research the use of ET in combination with MBT for SOA as described below:

- **Objective b.1) Test data generation using evolutionary techniques.** The applicant has a deep understanding of the test data for SOA and the associated challenges [4]. The test data provision is difficult in the area of information systems due to the multiple constraints between different fields of the desired data inputs and the constraints associated to the underlying business processes and system configurations. We will apply evolutionary and genetic-inspired algorithms for generating different mutants of the data that have as their evolution goal the satisfaction of the data constraints. This should provide a viable alternative to the test generation methods based on either constraint satisfaction which have an NP-complete computational complexity or model-checking algorithm that face the state space explosion problem. Moreover, the current techniques for test data generation are developed mainly for parameters having simple data types, which is certainly not the case for SOA testing. ET is versatile enough to work on tree-like data structures which thus can manipulate the complex input parameters required by the services.

- **Objective b.2) Investigating new combinations of ET and MBT for SOA**. The applicant published already an MBT method for service choreographies [5]. This was a first approach applying some brute force randomized algorithms (implemented by the co-author from IBM) for the search in the state space of the input test data. In this objective, we will adapt existing ET techniques to be able to tackle the new state-based SOA models describing the service interactions. Consequently new search-based techniques for testing communicating models such as the service choreographies will be developed to face additional testing problems that appear in the service integration phase. Note that ET for state-based specification is a recent research area. We will combine ET with MBT such that different state or transition coverage criteria for choreographies are achieved by exercising the corresponding guards and triggers of the service calls. Meta-heuristic search techniques are not standalone algorithms in themselves, but rather strategies ready for adoption to specific problems. We will refine such strategies to accommodate the specific elements of service choreographies like the different message communication semantics (send, receive or observe semantics) or the consistency of the local and global viewpoints.
- **Objective b.3) Enabling the transition from design time to runtime testing.** One of the characteristics of service-based systems is the clear shift of focus from design time to runtime, with the associated challenges. New classes of faults visible only during runtime must be addressed like late binding, composition faults or service agreement violations. In runtime testing, generation and running of the tests are dove-tailed together. The advantage is that test generation can take use the actual response of the system before computing the next test step and that it can benefit from the data obtained by monitoring the services. We envisage at least two ways in which ET could help. First is to intelligently direct the search through the large databases obtained from the service monitoring to extract suitable test data (this is essential given the difficulty of modeling and generating "good" test data at runtime). Second, given its native concepts of adaptation and mutation, ET could improve the MBT algorithms allowing for their adaptation and transformation during runtime to react timely and consistently to the runtime adaptation of the service-based system under test.

(c) Industrial application of the research methods

- **Objective c.1) Definition of an exploratory use case for fine tuning of**

the above methods. In order to properly validate and conduct the experiments associated with the above research, we will define an exploratory use case [6]. This will imply: (1) setting up of the testing framework implementing the MBT and ET algorithms and (2) definition of an academic use case of a service-based system including both test models as well as an implementation of the service-based systems. The concrete use case will be a simplified scenario like Purchase Order – Sales Order (taken for instance from open source benchmarks from the Web Service Test Forum - http://www.wstf.org/docs/scenarios) together with an open-source implementation of the scenario using for instance the Open Services Gateway Initiative (OSGi) framework. The testing framework will be based on the Eclipse platform that provides a rich library of modeling and testing plug-ins and allows a smooth integration of new implementations. Since the evolutionary techniques are generic techniques that must be tuned to the problem under consideration (in our case MBT and SOA testing domains), we intend to perform both a theoretical analysis of the problems as well as an experimental measurement approach, in order to find the best settings of the search parameters.

- **Objective c.2) Adapting the developed methods to Enterprise SOA domain.** One of the biggest problems facing software engineers and testers is that of scalability of results. Many approaches that are attractive and elegant in the laboratory turn out to be inapplicable in the field, because they lack scalability. The project will take into account from the beginning the scalability problem, using simulation and experimentation not only on the academic use case from objective c.1, but also on benchmarks from software industry. The goal domain will be Enterprise SOA. More concretely, **it will build upon the deep experience of the applicant gained during his postdoctoral research in the SAP Research lab over the last 3 years.** The applicant knows up to the smallest details the SOA platform implemented at SAP and is aware of the day-to-day experience of SAP testers from the dozens of hours of face-to-face meetings. Moreover, several requirements in the SOA testing, emerged from the MODELPLEX or DEPLOY industrial partners (like Telefonica, Thales, Siemens, Bosch) will be considered, e.g. from the area of Internet of Things. Since some of the modeling is done via DSLs (for choreography or GUI), the application of the research result will be enabled by model transformations to generic languages like UML.

Research methodology

The proposed project targets can be divided into theoretical, experimental and simulation, which are three methodologically distinct areas. However, the methods of research will follow the guidelines recommended for theoretical computer science [7], software engineering [8] and empirical software engineering [9,6]. The objectives a.1-a.2 and b.1-b.3, for example proposing SOA test modelling formalisms suitable for encoding into evolutionary algorithms, fall under theoretical computer science domain. The methods used in this case will involve: study of different formal models (UML or WS-*), defining the test concepts within their metamodels, proving the properties of the concepts by deductive methods. The theoretical results will be judged by the insights they reveal about the mathematical nature of these models and by their utility to the practice of computing and their ease of application. The project objectives belong to software engineering and the methods employed will consist in performing experiments that will follow the well-known templates provided in [6,9].

Bibliography:
[1] A. Stefanescu, M.F. Wendland, S. Wieczorek. "Using the UML testing profile standard for testing of enterprise service choreographies." *Submitted* to the European Conference on modeling foundations and applications (ECMFA'10), Springer proceedings, 2010.
[2] S. Wieczorek, A. Roth, A. Stefanescu, A. Charfi. "Precise steps for choreography modeling for SOA validation and verification." In Proc. of International Symposium on Service-Oriented System Engineering (SOSE'08), pp.148-153. IEEE Computer Society, 2008.
[3] S. Wieczorek, A. Stefanescu. "Improving Testing of Enterprise Systems by Model-based Testing on Graphical User Interfaces." To appear in the Proc. of the 7th Workshop on System Testing and Validation (STV'10), IEEE Computer Society, 2010.
[4] S. Wieczorek, A. Stefanescu, I. Schieferdecker. "Test data provision for ERP systems." In Proc. of the International Conference on Software Testing, Verification, and Validation (ICST'08), pp. 396-403. IEEE Computer Society, 2008.
[5] A. Stefanescu, S. Wieczorek, A. Kirshin. "MBT4Chor: a model-based testing approach for service choreographies." In Proc. of 5th European Conference on Model-Driven Architecture Foundations and Applications (ECMDA'09), LNCS, volume 5562, pp. 313-324. Springer, 2009.
[6] R. K. Yin. "Case Study Research, Design and Methods." Sage Publications, 2nd edition, 1994.
[7] G. Dodig-Crnkovic. "Scientific Methods in Computer Science.", Conference for the Promotion of Research in IT at New Universities and at University Colleges in Sweden, Skövde, 2002.
[8] C. B. Seaman. "Qualitative Methods in Empirical Studies of Software Engineering" IEEE Transactions on Software Engineering, Vol. 25, No. 4, pp.557-572, 1999.
[9] C. Wohlin, P. Runeson, M. Host, M.C. Ohlsson, B. Regnell, A. Wesslen. "Experimentation in Software Engineering - An Introduction." Kluwer, 2000.

Degree of originality/novelty/innovation and expected impact of the project

Originality of the proposed research

The proposed project activates in the highly active research domain of SOA modeling and testing, where there are many open problems and challenges as hinted in Section 8.2. There are new standards for SOA introduced only in the last years and the trend will certainly continue. All these models and standards must be complemented with a consistent perspective from testing point view (see Objectives a.1-a.2). The proposed solutions are based on two also recently successful areas, MBT and ET. They are to be combined in a different way than their current applications; we are not aware of any work doing this until now.

Moreover, the special requirements of SOA testing, including runtime testing, constitute a new domain where clear advancements can be made using MBT and ET both separately and in combination (see Objectives b.1-b.3).

Complexity of the research problems
The SOA domain brings in a new dimension of complexity on top of the known difficulties already known from the software engineering. The SOA systems are usually large systems covering not only activities of a single enterprise, but also the inter-organizational integration. For example, the Enterprise SOA developed in a model-driven way at SAP has a dozen types of models containing modeling information of business objects, deployment units, service components, service interfaces, integration scenarios, business process variants, service choreographies. It contains several thousands of services and several hundreds of choreographies and several millions of lines of code implementing them. Testing such a huge software system involves thousands of testers using several types of testing. It is essential that powerful automation tools and techniques are used to address the sheer complexity of the system. MBT is an ideal candidate to use the high-level information from the SOA models for test generation. ET brings in the smart algorithms of providing the right test data to make the generated test executable. However, to do this in the most efficient way requires innovative research and expertise. The proposed project aims to attack this motivating and complex problem.

Expected impact
Firstly, European Commission evaluated service-oriented architectures and service engineering as an the area of high potential innovation and impact on the European information society. A confirmation for that is the large amount of funding made available under FP7-ICT Programme for Research and Development Objective 1.2 "Internet of Services, Software and Virtualisation" and "Service and Software Architectures, Infrastructures and Engineering". The last FP7 call 5 will provide 90 million euros for the next wave of projects in Objective 1.2. On the one hand, **our proposed project will directly contribute to the European vision on “Internet of Services”** and on the other hand will profit from the latest advancement over the next 2 years in the area. Moreover, evolutionary testing was found also interesting enough by the European Commission with the effect of funding a whole research project on the topic: EvoTest (Evolutionary Testing for Complex Systems - http://evotest.iti.upv.es). EvoTest does not touch on SOA topics, but **our project will search for new applications of ET for SOA.**

Secondly, there is a **clearly signaled interest from the industry on the topic of the proposed project.** As we all know, there is a significant difference between *invention* and *innovation*, with innovation being a sum of inventions plus their successful application in practice. (Put in business metaphorical terms: “the invention is the conversion of cash into ideas, while innovation is the conversion of ideas into cash”). Based on this, we have our research driven by the practical application. For that, as mentioned also in Objective c.2, we will keep in touch with the practitioners. **More precisely, the applicant will continue to closely collaborate with its colleagues from SAP Research and apply the research on SAP examples.** The applicant was responsible inside SAP over the last years with the promotion of MBT for SOA inside the company. The topic will continue to be researched by SAP over the next years as well by different projects, namely the FP7 project DEPLOY will run until 2012 and one of the priority topics is MBT for SOA. Furthermore, SAP Research will start participating into a large project "Software cluster" (funded by the German Government, see http://www.ideen-zuenden.de/de/468.php), in which the SOA quality is among the investigated topics. Even more, the European Institute of Innovation and Technology (EIT) just recently announced the funding of a large EIT ICT lab with the topic of Future Internet (see http://www.eitictlabs.eu). SAP, Fraunhofer Institute and Technical University of Helsinki are part of the consortium and the applicant collaborated already with persons involved in this project. Last but not least, the applicant collaborated also with the group from IBM Research Haifa on the MBT topic and will continue to do so during the project. To conclude, **all the activities of the project will be fully aligned with these projects and contribute to the generated impact.**

Feasibility of the project (possible risks, alternative solutions):

The feasibility and credibility of the project are supported by the following essential factors:

- the applicant is an expert in the topics proposed in the project, with experience accumulated at top European universities (Munich, Edinburgh, Konstanz, Bucharest) and SAP Research lab, with many papers published over the last years in the area
- the strong group of research working from the University of Pitesti, Prof. Florentin Ipate, Prof. Tudor Balanescu, and their younger collaborators. Given the number of publications, their impact and visibility, the **software testing group from Pitesti is currently the strongest in Romania**. Prof. Ipate is coordinating a project in the timeframe 2008-2011 with the title "*An integrated evolutionary approach to formal modelling and testing*" funded by CNCSIS within the framework PN-II-ID-PCE-2008-2. This fits very well with our project on SOA.
- University of Pitesti has a very strong relation with group of Testing and Verification at the University of Sheffiled, led by Prof. Marian Gheorghe. This is a leading European group on testing, especially evolutionary testing and MBT. Prof. Gheorghe is also an Associate Professor (on leave) at the University of Pitesti.
- international connections established by the applicant at SAP Research, TU Munich (Prof. Javier Esparza), TU Berlin (Prof. Ina Schieferdecker), Univ. of Edinburgh (Prof. Colin Stirling), University of Boston (Prof. Calin Belta), Université de Franche-Comté (Prof. Bruno Legeard), University of Konstanz (Prof. Stefan Leue) and many others.

Possible risks and alternative solutions

As in any intrepid project, there are possible risks to be addressed:

- **Risk for Objective (a):** The combination of UML profiles for SOA and testing is artificial and inefficient.

  Mitigation: The applicant will discuss the problems encountered with the top experts in the areas, like Prof. Ina Schieferdecker, co-author of a Springer book on UML testing profiles and co-author of the U2TP standard. If difficulties still persist, the challenges will be documented in a conference paper to raise the awareness on the issue in the community.

- **Risk for Objective (b):** The developed evolutionary approaches are not very efficient for MBT SOA testing.

  Mitigation: Again, the experts in the domain (Prof. Ipate, Prof. Gheorghe, Prof. Legeard) will be approached to understand if this is a fundamental issue or just lack of experience of the applicant in the ET domain. Moreover, even if the applicant did not publish yet in the ET area, he has experience with similar techniques with search-based principles like directed model-checking (collaboration with Prof. Leue) and synthesis of robot control strategies (collaboration with Prof. Belta). Furthermore, there are alternative solutions from the area of metaheuristic search techniques like particle swarm optimization or simulated annealing.

- **Risk for Objective (c):** The research results do not scale on practical examples.

  Mitigation: The applicant has a lot of experience in industrial research. The validation on real examples will be done from the very beginning in the project (signing non-disclosure agreements with SAP). Moreover, the applicant has experience with several different MBT tools like Test Derivator from IBM Research, SpecExplorer from Microsoft, or TestDesigner from Smartesting designed by Prof. Legeard.

- **Risk for the whole project:** There are too many objectives to be addressed within the 2 year timeframe.

  Mitigation: Although it appears to be many things to be addressed, first there is the large experience of the applicant on the studied topics and second, there is a close connection between the different objectives, making them suitable to be studied together.